Congestion Control for Multicast Flows with Network Coding

Lijun Chen, *Member, IEEE*, Tracey Ho, *Member, IEEE*, Mung Chiang, *Fellow, IEEE*, Steven H. Low, *Fellow, IEEE*, and John C. Doyle

***Abstract*— Recent advances in network coding have shown great potential for efficient information multicasting in communication networks, in terms of both network throughput and network management. In this paper, we address the problem of flow control at end-systems for network coding based multicast flows. We formulate optimization based models for network resource allocation, based on which we develop two sets of decentralized controllers at sources and links/nodes for congestion control in wired networks with given coding subgraphs and without given coding subgraphs, respectively. With random network coding, both sets of controllers can be implemented in a distributed manner, and work at the transport layer to adjust source rates and at network layer to carry out network coding. We prove the convergence of the proposed controllers to the desired equilibrium operating points, and provide numerical examples to complement our theoretical analysis. The extension to wireless networks is also briefly discussed.**

***Index Terms*— Congestion control, Network coding, Multicast, Coding subgraph, Distributed algorithm.**

I. Introduction

Network coding extends the functionality of network nodes from storing/forwarding packets to performing algebraic operations on received data. Starting with the work of [1], which shows that employing coding at intermediate nodes is sometimes needed to maximize multicast throughput, various potential benefits of network coding have been shown, including robustness to link/node failures [15] and packet losses [6], [19]. Distributed random linear coding schemes, see, e.g., [5], [9], have made practical implementation of network coding possible. In this paper, we address the problem of flow control at end-systems for network coding based multicast flows with elastic rate demand.

Most existing work on network coding considers coding among packets of each multicast session, and assumes that the communication rates for each session and/or the network link capacities are fixed and known. Given a cost function in terms of the flow on each link, a min-cost flow optimization problem is obtained and solved to find the optimal coding subgraphs, which specify how much of each session's data should be sent on each link, see, e.g., [20], [28], [31]. For this reason, we call coding subgraphs of this kind *capacitated* subgraphs.

However, in many practical networks, traffic is bursty and elastic with varying rates, and since the network is shared by many users with unknown or changing demands, the available link capacities are unknown and variable. In such cases, it is not practical to solve a min-cost flow optimization to obtain capacitated subgraphs. Instead, congestion control is needed to make full use of bandwidth while avoiding congestion and maintaining certain fairness among the competing flows in the network.

One approach we propose is to use coding subgraphs that are *un-capacitated* (i.e., specifying which links are used by a session but not the amount of the data sent on each link) and chosen based on general cost criteria that are independent of flow rates. This is a practical approach; most existing routing approaches, such as those used in the Internet, specify analogously un-capacitated routes. Since each session uses only a limited set of trees, this approach may give lower rates compared to optimizing over the entire network, but it is much less complex. We propose decentralized controllers that combine congestion control at fast timescales and adaptive traffic splitting at slower timescales based on end-to-end congestion feedback in the network.

Another approach we consider does not explicitly find coding subgraphs, but makes dynamic routing and coding decisions based on queue length gradients. This approach, termed *back-pressure*, was first proposed for optimal routing and scheduling in [26] and extended to various contexts (see, e.g., [21]) including network coding in [11]. Our contribution in this part of the paper is to propose an alternative algorithm for back-pressure based routing and incorporate congestion control with network coding.

Our consideration of congestion control uses the framework of utility maximization, which can provide the flexibility of modeling user application needs or performance objectives and guide the design of distributed algorithms and decentralized control. As shown in, e.g., [13], [18], [16], TCP congestion control algorithms can be interpreted as distributed primal-dual algorithms over the Internet to maximize aggregate utility. We extend the basic utility maximization formulation to incorporate the two network coding approaches described above, and propose two sets of decentralized controllers for congestion control to meet the new challenges associated with network

Paper approved by Prof. Randall Berry. Manuscript received October 7, 2008; accepted May 9, 2012.

This work is partially supported by NSF through grants CNS-0435520, CNS-0520349, and CNS-0911041, DARPA through grant N66001-06-C-2020, the Caltech Lee Center for Advanced Networking, and Microsoft Research. This paper has been presented in part at the IEEE Conference on Computer Communications (Infocom), Anchorage, Alaska, USA, May 2007.

L. Chen is with the College of Engineering and Applied Science, University of Colorado, Boulder, CO 80309, USA (Email: lijun.chen@colorado.edu).

T. Ho, S. H. Low, and J. C. Doyle are with the Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA (Email: {tho, slow, doyle}@caltech.edu).

M. Chiang is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (Email: chiangm@princeton.edu).

coding. With random network coding, both sets of controllers can be implemented in a distributed manner, and work at the transport layer to adjust source rates and at the network layer to carry out network coding. We prove the convergence of the proposed controllers to the globally optimal solutions for intra-session network coding.

The main contribution of this paper is to present optimization models and propose decentralized congestion controllers for network coding based multicast flows. The proposed controllers are promising in practical implementation, and can be extended to handle different environments such as multi-layer network coding and multirate multicasting. In addition, in deriving decentralized control, we develop an alternative distributed algorithm – a partially-primal and dual gradient algorithm that, though presented for the specific problem we consider in this paper, is applicable to a certain class of nonstrict convex optimization problems. This algorithm is expected to find interesting applications in optimization and its application to engineering design and control.

The paper is organized as follows. The next section briefly discusses some related work. Section III presents details of the system model for multicast with intra-session network coding in wired networks. Sections IV and V present decentralized congestion controllers for multicast with and without given coding subgraphs, respectively. Section VI provides numerical examples to complement the theoretical analysis. Section VII briefly discusses the extension to wireless networks, and section VIII concludes with some discussions on further research.

II. Related work

There are several recent works on congestion control of multicast flows, see, e.g., [12], [7], [24], which consider traditional routing-based multicasting. In contrast, this paper studies congestion control for network coding based multicasting.

With network coding, the works most similar to our work are [20], [31], [29], [30]. We use a similar model but without network link costs for the networks without given coding subgraphs, see subsection III-C. What differentiates this part of our work from others are the following. First, we use a different decomposition and obtain a dynamic scheme that uses only local information, see section V. As an important consequence of such alternative decomposition, our solution requires less communication overhead. Our solution can also be readily extended to the case with network cost. Second, our congestion control scheme is a dual congestion control whose dual variables admit concrete and meaningful interpretation as congestion prices. Third, our work also differs from [20], [31] in that we do not relax the constraint that specifies the relation between the information flows and physical flows of a multicast session but exploit it to specify coding.

All existing work on network coding solves for the optimal coding subgraphs based on a flow model that is similar to multicommodity flow model for routing [8]. However, as discussed in the Introduction, it is often impractical to do so. In analogy to what happens with routing, we consider the case where subgraphs are chosen based on general cost criteria. We thus study congestion control for networks with given coding subgraphs, see sections III-B and IV.

Related work includes [23] that studies congestion control with adaptive multipath routing using a multi-commodity model for the routing. Our model for networks without given coding subgraphs is also a multi-commodity model but with the additional constraints from network coding, and moreover, we propose a different solution approach. For the case with given coding subgraphs, we use a technique similar to that from [8], [23] for adaptive control of traffic splits among different multicast trees. Related work also includes [22], [3] that studies flow control with backpressure based routing/scheduling. Our solution for networks without given coding subgraphs can be seen as an extension of those in [22], [3] to network coding.

III. Models and problem formulations

A. Network and Coding Model

Consider a network, denoted by a graph $\mathcal{G} = (N, L)$, with a set N of nodes and a set L of directed links. We denote a link either by a single index l or by the directed pair (i, j) of nodes it connects. Each link l has a fixed finite capacity c_l packets per second.

Let M denote the set of multicast sessions, indexed by m. Each session m has one source $s_m \in N$[1] and a set $D_m \subset N$ of destinations. Network coding allows flows for different destinations of a multicast session to share network capacity by being coded together: for a single multicast session m of rate x^m, information must flow at rate x^m to each destination; with coding the actual physical flow on each link need only be the maximum of the individual destination's flows [1]. These constraints can be expressed as

$$\sum_{j:(i,j)\in L} g_{i,j}^{md} - \sum_{j:(j,i)\in L} g_{j,i}^{md} = \begin{cases} x^m & \text{if } i = s_m \\ -x^m & \text{if } i = d \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$
$$\forall\, d \in D_m,$$

$$\max_{d\in D_m} \{g_{i,j}^{md}\} \le f_{i,j}^m, \ \forall\ (i,j) \in L, \quad (2)$$

where for each link (i, j), $g_{i,j}^{md}$ gives the *information flow* for destination d of session m, and $f_{i,j}^m$ gives the amount of link capacity that is allocated to session m. Note that the information flow balance equation (1) is formally similar to the physical flow balance equation for routing of data flows in the network. The inequality (2) simply says that the *physical flow* $g_{i,j}^m := \max_{d\in D_m}\{g_{i,j}^{md}\}$ for each session m should not exceed its allocated link capacity.

Figure 1 gives an example, adapted from [1], of a linear network code, and the corresponding flow variables $(g_{i,j}, g_{i,j}^{d_1}, g_{i,j}^{d_2})$. For packet networks, the result is stated formally in Theorem 1 of [20], which we reproduce here, slightly adapted:

Theorem 1: The rate vector g satisfies the constraints (1) if and only if there exists a network code that sets up a multicast connection at rate arbitrarily close to x^m from source

[1]Our analysis can extend to handle multi-source multicasting in a straightforward way.

s_m to destinations in set D_m and that injects packets at rate arbitrarily close to $g^m_{i,j}$ on each link (i,j).

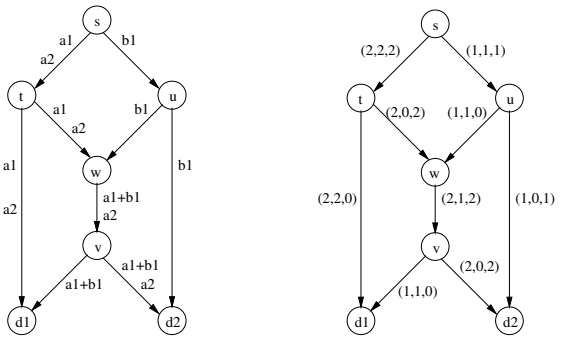


Fig. 1. An example network coding subgraph with one source s and two destinations d_1 and d_2 (left graph), where links (s,u), (u,w) and (u,d_2) are assumed to have one unit of capacity, and all other links have two units of capacity; and the corresponding flow variables (right graph), where each link (i,j) is marked by the triple $(g_{i,j}, g^{d_1}_{i,j}, g^{d_2}_{i,j})$.

For the case of multiple sessions sharing a network, achieving optimal throughput requires in some cases coding across sessions. However, designing such codes is a complex and largely open problem. Thus, we limit our consideration to separate network codes operating within each session, an approach referred to as superposition coding [32] or intra-session coding. In this case, the set of feasible flow vectors is specified by combining constraints (1)-(2) for each session $m \in M$ with the following link capacity constraints:

$$\sum_{m \in M} f^m_{i,j} = c_{i,j} \ , \quad \forall\, (i,j) \in L. \tag{3}$$

In practice, the network codes can be designed using the approach of distributed random linear network coding, see, e.g., [9], [5], in which network nodes form output packets by taking random linear combinations of corresponding blocks of bits in input packets. The linear combination corresponding to each packet can be specified by a coefficient vector in the packet header, updated by applying to the coefficient vectors the same linear transformations as to the data. If (1)-(2) holds, each sink receives with high probability a set of packets with linearly independent coefficient vectors, allowing it to decode. The relative overhead of these coefficient vectors depends on parameters of the network code that can be chosen to trade-off overhead against performance, and it decreases with the size of the packets. See, e.g., [5], [11] for a detailed description and discussion of overhead and other practical implementation issues.

B. Multicast with Given Coding Subgraphs

We first consider the network with a given coding subgraph[2] G_m for each session m. The subgraph G_m can be viewed as the union of links of a set R_m of possibly overlapping multicast trees, each connecting source s_m to all destinations $d \in D_m$. Congestion control is carried out by adjusting the flow rate on each tree. Coding is done on overlapping segments of different trees of a session that have disjoint sets of downstream destinations. Figure 2 shows an example of muticast trees that are decomposed from the coding subgraph shown in Figure 1. In this example, coding on the shared link is possible, allowing both trees to simultaneously transmit information at their maximum individual rates.

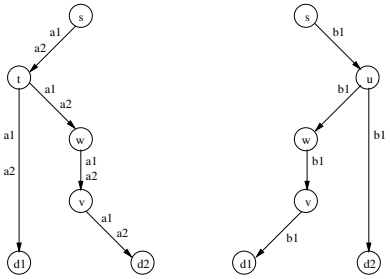


Fig. 2. Multicast trees for the example shown in Fig.1. Coding is done on the shared link (w,v), which, as part of the left tree, has one downstream destination d_2, and, as part of the right tree, has one downstream destination d_1. The left tree can support up to two units of information flow and the right tree can independently support up to one unit of information flow, since coding on link (w,v) allows the two trees to share capacity.

Analogous to practical routing, such coding subgraphs can be chosen in a variety of ways based on combinations of different considerations, such as delay, resource usage or commercial relationships among network providers. For instance, we can use existing multicast tree construction algorithms, or use existing techniques for finding multiple paths to each destination and combine appropriate sets of paths that form trees.

To simplify notation, we consider the case where overlapping segments of different trees of a session have disjoint sets of downstream destinations, thus allowing coding to occur on all overlapping segments[3]; the more general case where coding occurs only on some overlapping segments admits a similar analysis. Each tree T^m_r, $r \in R_m$ contains a set $L_r \subset L$ of links, which defines a $|L| \times |R_m|$ multicast matrix H^m whose (l,r)th entry is given by

$$H^m_{lr} = \begin{cases} 1 & \text{if } \ l \in L_r \\ 0 & \text{otherwise.} \end{cases}$$

Note that over each multicast tree T^m_r the source sends the same information flow to each destination; we denote its rate by x^m_r. With intra-session network coding, the physical flow rate for each multicast session m though link l is $\max_r\{H^m_{lr} x^m_r\}$. For each link l, denote by y^m_l the amount of link capacity that is allocated to session m. The link capacity constraints (2)-(3) become

$$\max_r\{H^m_{lr} x^m_r\} \le y^m_l, \tag{4}$$

$$\sum_m y^m_l = c_l \ \ \forall\, l \in L. \tag{5}$$

By Theorem 1, conditions (4)-(5) are satisfied if and only if there exists a corresponding multicast network code of rate

[2]In this and the following sections, subgraph refers to "un-capacitated" subgraph.

[3]This is the case, for instance, if each session's trees have been formed by first finding multiple link-disjoint paths to each destination and then choosing combinations of these paths that form trees.

arbitrarily close to $\sum_r x_r^m$ from source s_m to destinations $d \in D_m$.

Following [13], assume each session m attains a utility $U_m(x^m)$ when it transmits at a rate $x^m = \sum_r x_r^m$ packets per second over the coding subgraph. We assume $U_m(\cdot)$ is continuously differentiable, increasing, and strictly concave for the flows with elastic rate demand. Our objective is to design decentralized controllers at sources and links/nodes to achieve the optimum of the following network resource allocation problem:

$$\begin{aligned}\mathbf{P1}: \quad \max_{x_r^m, y_l^m} \quad & \sum_m U_m(x^m) \\ \text{subject to} \quad & H_{lr}^m x_r^m \le y_l^m, \ \forall r \in R_m, \ \forall m \in M \\ & \sum_m y_l^m = c_l, \ \forall l \in L.\end{aligned}$$

C. Multicast without Given Coding Subgraphs

Since coding subgraphs are not given, we directly use the network coding flow constraints (1)-(3) and Theorem 1, given in subsection III-A, to formulate the following optimization problem which chooses source rates x^m, information rates $g_{i,j}^{md}$ and link capacity allocation $f_{i,j}^m$ so as to maximize aggregate utility:

$$\begin{aligned}\mathbf{P2}: \quad \max_{x,g,f} \quad & \sum_m U_m(x^m) \\ \text{subject to} \quad & \sum_{j:(i,j)\in L} g_{i,j}^{md} - \sum_{j:(j,i)\in L} g_{j,i}^{md} = x_i^m, i \ne d, \forall d, m \\ & g_{i,j}^{md} \le f_{i,j}^m, \ \forall d, m \\ & \sum_m f_{i,j}^m = c_{i,j}, \ \forall (i,j) \in L,\end{aligned}$$

where $x_i^m = x^m$ if $i = s_m$ and $x_i^m = 0$ otherwise. Here we do not include flow balance equation at destinations, which is automatically guaranteed by the flow balance at the source and intermediate nodes.

Note that in the models **P1** and **P2**, network coding comes into action through the constraints (4) and (2). With Theorem 1, this gives some form of "separation principle" that allows us to separate decisions on resource usage and congestion control from the design of the actual network codes.

The system problems **P1** and **P2** are convex optimization problems, and are polynomially solvable if all the utilities and constraint information is provided, but this is impractical in real networks. Since they are convex optimization problems with strong duality, distributed algorithms and decentralized control can be derived by considering corresponding Lagrange dual problems, as we will show in the next two sections.

IV. Decentralized congestion control for Networks with Given Coding Subgraphs

We introduce for each multicast session m traffic split variables $\alpha_r^m \ge 0$ for each multicast tree T_r^m of the coding subgraph, such that $\sum_r \alpha_r^m = 1$ and $x_r^m = x^m \alpha_r^m$. We see that α_r^m controls the fraction of the traffic of multicast session m that is sent through the tree T_r^m. Instead of solving the problem **P1** directly, we first consider the version of the rate control problem with the fixed split vector α.

$$\begin{aligned}\mathbf{P1a}: \quad \max_{\{x^m, y_l^m\}} \quad & \sum_m U_m(x^m) \\ \text{subject to} \quad & H_{lr}^m x^m \alpha_r^m \le y_l^m \\ & \sum_m y_l^m = c_l.\end{aligned}$$

The above problem is a strictly convex and has a unique solution, with respect to source rates x^m. Let us denote its maximum by $U(\alpha)$. The system problem **P1** corresponds to computing

$$\begin{aligned}\mathbf{P1b}: \quad \max_{\alpha \ge 0} \quad & U(\alpha) \\ \text{subject to} \quad & \sum_r \alpha_r^m = 1.\end{aligned}$$

Note that the above problem is not necessarily convex. But we will see later that it can still be solved for global optimality.

A. Two-Timescale Flow Control

Consider the Lagrangian of the problem **P1a** with respect to the constraints due to network coding

$$L(\alpha, p, x, y) = \sum_m U_m(x^m) - \sum_{l,m,r} p_{l,r}^m (H_{lr}^m x^m \alpha_r^m - y_l^m).$$

Interpreting $p_{l,r}^m$ as the "congestion price" at link l for multicast tree T_r^m, and motivated by maximizing the Lagrangian over x and y for fixed p, we obtain the following joint congestion control and session allocation algorithm:

Congestion control: Given congestion price p, the source s_m adjusts flow rate x^m according to the aggregate congestion price $\sum_l H_{l,r}^m p_{l,r}^m$ over the multicast trees T_r^m,

$$x^m = (U_m')^{-1}\Big(\sum_r \alpha_r^m \sum_l H_{l,r}^m p_{l,r}^m\Big). \tag{6}$$

Similar to TCP congestion control algorithm where the source adjusts its sending rate according to aggregate congestion price along its path, this congestion control mechanism has the desired price structure and is an end-to-end congestion control mechanism.

Session allocation: Intuitively, the multicast session with higher link price should be allocated more link capacity. Let $p_l^m = \sum_r p_{l,r}^m$ and denote by $\eta_l[p(t)]$ the minimal of those $\bar{p}_l(t)$ at time t such that $\bar{p}_l(t) = \frac{1}{|M_l'(t)|}\sum_{m\in M_l'(t)} p_l^m(t)$ with $M_l'(t) := \{m | y_l^m(t) > 0 \text{ or } p_l^m(t) \ge \bar{p}_l(t), m \in M\}$.[4] At each link l, the amount of capacity y_l^m that is allocated to session m follows

$$\dot{y}_l^m = \varepsilon_l [p_l^m - \eta_l[p]]_{y_l^m}^+, \tag{7}$$

[4] $\bar{p}_l$ and M_l' can be determined in a recursive way as follows. In the beginning, let $M_l' = M$ and calculate $\bar{p}_l = \frac{1}{|M_l'|}\sum_{m\in M_l'} p_l^m(t)$, and then exclude from M_l' those sessions m such that $y_l^m = 0$ and $p_l^m < \bar{p}_l$. Repeat the same procedure with the new sets M_l', and when it stops we get $\eta_l[p]$. $\eta^m[p]$ and $\eta_{i,j}[w]$ that appear later can be determined in similar way.

where ε_l is a positive stepsize, and $[h]_z^+ = h$ if $z > 0$ and $[h]_z^+ = \max\{0, h\}$ if $z = 0$. It is easy to verify that

$$\begin{aligned} \sum_m \dot{y}_l^m &= 0, \\ \sum_m \dot{y}_l^m p_l^m &\geq 0. \end{aligned}$$

We see that $\sum_m \dot{y}_l^m p_l^m = 0$ only if $\dot{y}_l^m = 0$, which requires $p_l^m = \bar{p}_l$, or, $y_l^m = 0$ and $p_l^m < \bar{p}_l$.

The session allocation algorithm (7) actually follows the gradient direction of $\sum_m p_l^m y_l^m$ subject to $\sum_m y_l^m = c_l$. Any algorithms that follow the gradient directions would work, and (7) just picks a specific gradient direction that enables the convergence analysis.

Defining $D(\alpha, p) = \max_{x,y} L(\alpha, p, x, y)$ with $\sum_m y_l^m = c_l$, by duality we have (see, e.g., Chapter 5 in [2])

$$U(\alpha) = \min_{p \geq 0} D(\alpha, p) = \min_{p \geq 0} \max_{x,y} L(\alpha, p, x, y).$$

The dual problem $\min_p D(\alpha, p)$ can be solved by using the gradient method [2], where Lagrangian multipliers are adjusted in the opposite direction to the gradient $\partial_p D(\alpha, p)$. This motivates the following dual congestion price update mechanism.

Congestion price update: Link l price with respect to multicast tree T_r^m follows

$$\dot{p}_{l,r}^m = \gamma_l [H_{lr}^m \alpha_r^m x^m(p) - y_l^m(p)]_{p_{l,r}^m}^+, \tag{8}$$

where γ_l is a positive stepsize. Note that link l will use capacity y_l^m to transfer coded packets for multicast session m, equation (8) says that if the demand $H_{lr}^m x_r^m$ for virtual capacity at link l for the information flow of multicast tree T_r^m exceeds the assigned physical capacity y_l^m, the price p_{lr}^m will rise, and decreases otherwise. Equation (8) is distributed and can be implemented at individual links using only local information.

Note that the usual way to solve the problem like **P1a** distributedly is to also relax the equality constraints like $\sum_m y_l^m = c_l$. We want to avoid this, since it will introduce auxiliary control variables that are unnecessary and do not admit physical interpretation. We also do not maximize the linear term $\sum_m p_l^m y_l^m$ in the Lagrangian directly, since this will lead to oscillations and nonsmoothness. The distributed algorithm (6)-(8) is a partially-primal and dual gradient algorithm. Its convergence analysis is subtle, due to the equality constraints. To our knowledge, the specific gradient direction we choose in equation (7) is the only gradient direction for which the global convergence has been analytically established, see the following convergence analysis. Though developed for the specific problem **P1a**, this algorithm and its convergence analysis are applicable to a class of optimization problems with similar structure.

The above congestion control algorithm (6)-(8) works under the assumption that the traffic split vector α remains constant. We now discuss how to control α_r^m to solve the problem **P1b**, which we call tree adaptation. We assume that tree adaptation is much slower so that the minimization of $D(\alpha, p)$ over p can be seen as instantaneous.

Intuitively, the optimal traffic split vector should strike an equilibrium that is similar to Wardrop equilibrium, where for each multicast session the aggregate prices in all multicast trees actually used are equal and less than those which would be experienced by a single packets on any unused tree [27]. We gradually update the split vector towards this equilibrium, as in [8]. Let $p_r^m = \sum_l H_{lr}^m p_{l,r}^m$ and denote by $\eta^m[p(t)]$ the maximal of those $\bar{p}^m(t)$ at time t such that $\bar{p}^m(t) = \frac{1}{|R'_m(t)|} \sum_{r \in R'_m(t)} p_r^m(t)$ with $R'_m(t) := \{r | \alpha_r^m(t) > 0 \text{ or } p_r^m(t) \leq \bar{p}^m(t), r \in R_m\}$.

Tree adaptation: Each source s_m controls the traffic split variable α_r^m following

$$\dot{\alpha}_r^m = \kappa_m [\eta^m[p] - p_r^m]_{\alpha_r^m}^+, \tag{9}$$

where κ_m is a positive stepsize. It is straightforward to verify that

$$\sum_r \dot{\alpha}_r^m = 0, \tag{10}$$

$$\sum_r \dot{\alpha}_r^m p_r^m \leq 0. \tag{11}$$

We see that $\sum_r \dot{\alpha}_r^m(n) p_r^m = 0$ only if $\dot{\alpha}_r^m(n) = 0$, which requires

$$p_r^m \geq \bar{p}^m, \tag{12}$$

$$\alpha_r^m (p_r^m - \bar{p}^m) = 0. \tag{13}$$

B. Convergence Analysis

The decentralized controllers for flow control presented in last subsection have embedded loops. In the inner loop (6)-(8), which operates at a fast timescale, the network searches for optimal source rates, session allocation and congestion prices for fixed flow split vector. In the outer loop (9), which operates at a slow, traffic engineering timescale, the sources adapt the flow split vector based on the stabilized congestion prices in the network. The tree adaptation algorithm (9) can be seen as a method for stable traffic engineering based on congestion prices.

We now provide the convergence analysis of the inner loop controllers (6)-(8). Denote by p^* an optimal solution to the dual problem $\min_p D(\alpha, p)$, and x^* and y^* the optimal source rate and session allocation of problem **P1a**. By the optimality conditions for convex program, we have

$$(x^*)^m = (U'_m)^{-1} \Big(\sum_r \alpha_r^m \sum_l H_{l,r}^m (p^*)_{l,r}^m \Big), \tag{14}$$

$$(y*)_l^m = \arg \max_{\sum_m y_l^m = c_l} \sum_{m,l} (p^*)_l^m y_l^m, \tag{15}$$

$$(p^*)_{l,r}^m (H_{lr}^m \alpha_r^m (x^*)^m - (y^*)_l^m) = 0, \ (p^*)_{l,r}^m \geq 0, \tag{16}$$

$$H_{lr}^m \alpha_r^m (x^*)^m \leq (y^*)_l^m, \ \sum_m (y^*)_l^m = c_l. \tag{17}$$

Theorem 2: Under congestion control and session allocation (6)-(8), the system converges to the optimum of the problem **P1a**.

Proof: Consider the Lyapunov function $V(p,y) = \sum_{m,l,r} \frac{(p^m_{l,r}-(p^*)^m_{l,r})^2}{2\gamma_l} + \sum_{m,l} \frac{(y^m_l-(y^*)^m_l)^2}{2\varepsilon_l}$. We have

$$\begin{aligned}
\dot{V}(p,y) &= \sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})[H^m_{lr}\alpha^m_r x^m - y^m_l]^+_{p^m_{l,r}} \\
&\quad + \sum_{m,l}(y^m_l-(y^*)^m_l)[p^m_l-\eta_l[p]]^+_{y^m_l} \\
&\le \sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})(H^m_{lr}\alpha^m_r x^m - y^m_l) \\
&\quad + \sum_{m,l}(y^m_l-(y^*)^m_l)(p^m_l-\eta_l[p]) \\
&= \sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})(H^m_{lr}\alpha^m_r x^m - y^m_l) \\
&\quad + \sum_{m,l}(y^m_l-(y^*)^m_l)p^m_l \\
&= \sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})(H^m_{lr}\alpha^m_r x^m - H^m_{lr}\alpha^m_r (x^*)^m) \\
&\quad + \sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})(H^m_{lr}\alpha^m_r (x^*)^m - (y^*)^m_l) \\
&\quad + \sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})((y^*)^m_l - y^m_l) \\
&\quad + \sum_{m,l}(y^m_l-(y^*)^m_l)p^m_l \\
&= \sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})(H^m_{lr}\alpha^m_r x^m - H^m_{lr}\alpha^m_r (x^*)^m) \\
&\quad + \sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})(H^m_{lr}\alpha^m_r (x^*)^m - (y^*)^m_l) \\
&\quad + \sum_{m,l}(y^m_l-(y^*)^m_l)(p^*)^m_l.
\end{aligned}$$

Since the marginal utility $U'_m(\cdot)$ is a decreasing function and so is its inverse, we have

$$\sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})(H^m_{lr}\alpha^m_r x^m - H^m_{lr}\alpha^m_r (x^*)^m) \le 0.$$

By the optimality conditions (16)-(17), we have

$$\sum_{m,l,r}(p^m_{l,r}-(p^*)^m_{l,r})(H^m_{lr}\alpha^m_r (x^*)^m - (y^*)^m_l) \le 0,$$

and by equation (15),

$$\sum_{m,l}(y^m_l-(y^*)^m_l)(p^*)^m_l \le 0.$$

Thus, $\dot{V}(p,y) \le 0$. This implies that the system converges to an invariant set specified by $\dot{V}(p,y) = 0$ [14]. Furthermore, $\dot{V}(p,y) = 0$ only if p, y and x satisfy the optimality conditions (14)-(17). ∎

Now we study the convergence of the outer loop algorithm.

Theorem 3: The tree adaptation algorithm (9) converges to the optimum of the system problem **P1**.

Proof: Note that

$$\begin{aligned}
U(\alpha) &= \min_p D(\alpha,p) \\
&= \min_p\{\sum_m U_m(x^m(p)) \\
&\quad - \sum_{m,l,r} p^m_{l,r}(H^m_{l,r}\alpha^m_r x^m(p) - y^m_l(p))\}. \quad (18)
\end{aligned}$$

So, the differential of $U(\alpha)$ can be written as

$$\begin{aligned}
dU(\alpha) &= \frac{\partial D(\alpha,p)}{\partial p}dp + \frac{\partial D(\alpha,p)}{\partial \alpha}d\alpha \\
&= \frac{\partial D(\alpha,p^*)}{\partial p}dp - \sum_{m,l,r} x^m H^m_{l,r}(p^*)^m_{l,r}d\alpha^m_r,
\end{aligned}$$

where $p^* = \arg\min_{p'} D(\alpha,p')$. Since p^* minimizes $D(\alpha,p)$ given α, $\frac{\partial D(\alpha,p^*)}{\partial p}$ cannot be a descent direction. So, $\frac{\partial D(\alpha,p^*)}{\partial p}dp \ge 0$. Hence,

$$dU(\alpha) \ge -\sum_{m,l,r} x^m H^m_{l,r}(p^*)^m_{l,r}d\alpha^m_r, \quad (19)$$

i.e.,

$$\dot{U}(\alpha) \ge -\sum_{m,l,r} x^m H^m_{l,r}(p^*)^m_{l,r}\dot{\alpha}^m_r. \quad (20)$$

By (11), we have $\dot{U}(\alpha) \ge 0$. So, the tree adaptation algorithm (9) will converge to an equilibrium α^* such that $\dot{U}(\alpha^*) = 0$. However, this only guarantees the convergence of the tree adaptation algorithm. Without further elaboration, we cannot even claim it solves for a local optimal of the problem **P1b**.

Note that, following equations (6) and (13), we obtain at $(\alpha^*, p(\alpha^*), x(\alpha^*))$

$$U'_m(x^m) = \frac{\partial U_m}{\partial x^m_r}(x^m) = \sum_l H^m_{l,r}p^m_{l,r}, \text{ if } x^m_r > 0, \quad (21)$$

$$\sum_l H^m_{l,r}p^m_{l,r} \ge U'_m(x^m), \text{ if } x^m_r = 0, \quad (22)$$

which means that

$$x(\alpha^*) = \arg\max_{x^m_r}\sum_m U_m(\sum_r x^m_r) - \sum_{m,r,l} p^m_{l,r}(\alpha^*)H^m_{l,r}x^m_r. \quad (23)$$

Also, we have $y(\alpha^*) = \arg\max_y \sum_{m,r,l} p^m_{l,r}(\alpha^*)y^m_l$. Denote the Lagrangian of the system problem **P1** with respect to the constraints due to network coding as $\hat{L}(p,x,y)$. We have

$$(x(\alpha^*), y(\alpha^*)) = \arg\max_{x,y}\hat{L}(p(\alpha^*),x,y). \quad (24)$$

Furthermore, by duality between the problem **P1a** and its dual, we have

$$\sum_{m,l,r} p^m_{l,r}(\alpha^*)(H^m_{l,r}x^m_r(\alpha^*) - y^m_l(\alpha^*)) = 0. \quad (25)$$

Combining (24)-(25), we conclude that

$$\sum_m U_m(\sum_r x^m_r(\alpha^*)) = \hat{L}(p(\alpha^*),x(\alpha^*),y(\alpha^*)), \quad (26)$$

which by duality only happens when $p(\alpha^*)$ and $x^m_r(\alpha^*)$ solve the system problem **P1** and its dual. So, the tree adaptation algorithm (9) indeed solves the system problem **P1**. This also

proves that the tree adaptation algorithm solves the problem **P1b**. ■

To establish the convergence of the tree adaptation algorithm, we have only used the property (10)-(11). The algorithm (9) is only one specific implementation of (10)-(11), and any algorithms that satisfy (10)-(11) would work. Also, note that the adaptation algorithm (9) and Theorem 3 can be readily extended to routing-based multicasting and multipath routing.

Remarks: We have proposed a two-timescale flow control for network coding based multicast flows with given coding subgraphs. The separation of timescales is a reasonable assumption, since in real networks it is not a sensible practice to do routing to avoid congestion at the timescale of congestion control. However, mathematically it would be nice if the above algorithm converges without the assumption of the separation of timescales. Similar to the proof of Theorem 2, we can establish the local stability of flow control (6)-(9) without the separation of timescales, by considering Lyapunov function $V(p,y,\alpha) = \sum_{m,l,r} \frac{(p^m_{l,r}-(p^*)^m_{l,r})^2}{2\gamma_l} + \sum_{m,l} \frac{(y^m_l-(y^*)^m_l)^2}{2\varepsilon_l} + \sum_{m,r} \frac{(x^*)^m(\alpha^m_r-(\alpha^*)^m_r)^2}{2\kappa_m}$.

C. Implementation of Price Feedback

Each link l keeps a separate virtual queue p^m_{lr} for each multicast tree T^m_r of each session m which acts as the congestion price. Each packet's header contains the indexes of the trees whose information it contains. When a packet is received at a node from an incoming link l, if the packet header contains the rth tree index, the queue size p^m_{lr} is increased by one; otherwise it is unchanged. Similarly, when a packet is sent by a node on an outgoing link l, if the packet header contains the rth tree index, the queue size p^m_{lr} is decreased by one; otherwise it is unchanged. The congestion prices over a multicast tree are fed back to the source node in the following way. Each node i in the tree will pass the aggregate price along the links from the receivers till itself to the upstream node j ("upstream" is defined as the direction from receivers to source node over a multicast tree). In this recursive way, the source node will get the aggregate congestion prices over that multicast tree, and adjust the sending rate accordingly.

V. Decentralized Congestion Control for Networks without Given Coding Subgraphs

A. Distributed Algorithm

Now we turn to system problem **P2** and consider its Lagrangian with respect to the flow balance constraints,

$$L(p,x,g,f) = \sum_m U_m(x^m) - \sum_{i,m,d\in D_m} p^{md}_i(x^m_i - \sum_{j:(i,j)\in L} g^{md}_{i,j} + \sum_{j:(j,i)\in L} g^{md}_{j,i}).$$

Interpreting p^{md}_i as the "congestion price" at node i for multicast session m and destination $d \in D_m$, and motivated by maximizing the Lagrangian over x, g and f for fixed p, we obtain the following joint congestion control and session allocation algorithm:

Congestion control: Given congestion price p, each source node s_m adjusts its sending rate according to local congestion price that is generated locally at the source node,

$$x_m = {U'_m}^{-1}(\sum_{d\in D_m} p^{md}_{s_m}). \quad (27)$$

Note that

$$\begin{aligned} &\max_{g,f} \sum_{i,m,d} p^{md}_i(\sum_j g^{md}_{i,j} - \sum_j g^{md}_{j,i}) \ \ \textit{s.t.} \ \ g^{md}_{i,j} \le f^m_{i,j} \\ =\ &\max_{g,f} \sum_{i,j,m,d} g^{md}_{i,j}(p^{md}_i - p^{md}_j) \ \ \textit{s.t.} \ \ g^{md}_{i,j} \le f^m_{i,j} \\ =\ &\max_{f} \sum_{i,j,m,d} f^m_{i,j}[p^{md}_i - p^{md}_j]^+, \end{aligned}$$

where '+ denotes the projection onto the set $\mathcal{R}^+$ of non-negative real numbers. Similarly to that in section IV, the session allocation algorithm should follow the gradient direction of $\sum_{i,j,m,d} f^m_{i,j}[p^{md}_i - p^{md}_j]^+$. Each node i collects congestion price information from its neighbor j, and calculates differential price $w^m_{i,j}(t) = \sum_d [p^{md}_i(t) - p^{md}_j(t)]^+$. Denote by $\eta_{i,j}[w(t)]$ the minimal of those $\bar{w}_{i,j}(t)$ at time t such that $\bar{w}_{i,j}(t) = \frac{1}{|M'_l(t)|}\sum_{m\in M'_l(t)} w^m_{i,j}(t)$ with $M'_l(t) := \{m | f^m_{i,j}(t) > 0 \text{ or } w^m_{i,j}(t) \ge \bar{w}_{i,j}(t), m \in M\}$.

Session allocation: At each link (i,j), the amount of capacity $f^m_{i,j}$ that is allocated to session m follows

$$\dot{f}^m_{i,j} = \varepsilon_{i,j}[w^m_{i,j} - \eta_{i,j}[w]]^+_{f^m_{i,j}}, \quad (28)$$

where $\varepsilon_{i,j}$ is a positive stepsize. Similarly, it is easy to verify that

$$\begin{aligned} \sum_m \dot{f}^m_{i,j} &= 0, \\ \sum_m \dot{f}^m_{i,j} w^m_{i,j} &\ge 0. \end{aligned}$$

We see that $\sum_m \dot{f}^m_{i,j} w^m_{i,j} = 0$ only if $\dot{f}^m_{i,j} = 0$, which requires $w^m_{i,j} = \bar{w}_{i,j}$, or, $f^m_{i,j} = 0$ and $w^m_{i,j} < \bar{w}_{i,j}$.

Over link (i,j), a random linear combination of data of multicast session m to all destinations d such that $p^{md}_i - p^{md}_j > 0$ is sent at rate $f^m_{i,j}$. Mathematically, this is equivalent to solving the primal variable g by the following assignment

$$g^{md}_{i,j} = \begin{cases} f^m_{i,j} & \text{if } p^{md}_i - p^{md}_j > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (29)$$

Define

$$\begin{aligned} D(p) = \max_{x_m, g^{md}_{i,j}, f^m_{i,j}} \quad & L(p,x,g,f) \\ \text{subject to} \quad & g^{md}_{i,j} \le f^m_{i,j}, \ \sum_m f^m_{i,j} = c_l. \end{aligned}$$

Again the dual problem $\min_p D(p)$ can be solved by using the gradient method. This motivates the following dual congestion price update mechanism.

Congestion price update: Node i price with respect to multicast session m and destination $d \in D_m$ follows

$$\dot{p}^{md}_i = \gamma_i[x^{md}_i(p) - \sum_{j:(i,j)\in L} g^{md}_{i,j}(p) + \sum_{j:(j,i)\in L} g^{md}_{j,i}(p)]^+_{p^{md}_i}, \quad (30)$$

where γ_i is a positive stepsize.

With the above controllers, each source node adjusts its sending rate according to the local congestion price. Thus, there is no communication overhead for congestion control. The majority of communication overhead is for session allocation, but that only requires nodes to communicate with direct neighbors. Thus, our design has very low communication overhead, compared with other schemes with similar models [20], [31], [29], [30]. Note that the above session allocation component uses *back-pressure* to do optimal routing and specify coding, similarly to [11]. Such dynamic network coding based multicasting offers both a larger rate region and much lower complexity, as compared to optimal dynamic routing based multicasting [24].

B. Convergence Analysis

The decentralized controllers (27)-(30) are also a partially-primal and dual gradient algorithm. Denote by x^*, g^*, f^* and p^* the optimal primal and dual variables for the problem **P2**. Similarly, we have the following convergence result.

Theorem 4: Under congestion control and session allocation(27)-(30), the system converges to the optimum of the problem **P2**.

Proof: Consider Lyapunov function $V(p,f) = \sum_{m,d,i} \frac{(p_i^{md}-(p^*)_i^{md})^2}{2\gamma_l} + \sum_{m,(i,j)} \frac{(f_{i,j}^m-(f^*)_{i,j}^m)^2}{2\varepsilon_{i,j}}$. We have

$$\begin{aligned}
&\dot{V}(p,f)\\
=&\sum_{m,d,i}(p_i^{md}-(p^*)_i^{md})[x_i^{md}(p)\\
&-\sum_{j:(i,j)\in L} g_{i,j}^{md}(p)+\sum_{j:(j,i)\in L} g_{j,i}^{md}(p)]^+_{p_i^{md}}\\
&+\sum_{m,(i,j)\in L}(f_{i,j}^m-(f^*)_{i,j}^m)[w_{i,j}^m-\eta_{i,j}[p]]^+_{f_{i,j}^m}\\
\le&\sum_{m,d,i}(p_i^{md}-(p^*)_i^{md})(x_i^{md}(p)\\
&-\sum_{j:(i,j)\in L} g_{i,j}^{md}(p)+\sum_{j:(j,i)\in L} g_{j,i}^{md}(p))\\
&+\sum_{m,(i,j)\in L}(f_{i,j}^m-(f^*)_{i,j}^m)(w_{i,j}^m-\eta_{i,j}[p])\\
=&\sum_{m,d,i}(p_i^{md}-(p^*)_i^{md})(x_i^{md}(p)-\sum_{j:(i,j)\in L} g_{i,j}^{md}(p)\\
&+\sum_{j:(j,i)\in L} g_{j,i}^{md}(p))+\sum_{m,(i,j)\in L}(f_{i,j}^m-(f^*)_{i,j}^m)w_{i,j}^m\\
=&\sum_{m,d,i}(p_i^{md}-(p^*)_i^{md})(x_i^{md}(p)-(x^*)_i^{md})\\
&+\sum_{m,d,i}(p_i^{md}-(p^*)_i^{md})(\sum_{j:(i,j)\in L}(g^*)_{i,j}^{md}(p)\\
&-\sum_{j:(j,i)\in L}(g^*)_{j,i}^{md}(p)-\sum_{j:(i,j)\in L} g_{i,j}^{md}(p)\\
&+\sum_{j:(j,i)\in L} g_{j,i}^{md}(p))+\sum_{m,d,i}(p_i^{md}-(p^*)_i^{md})((x^*)_i^{md}(p)\\
&-\sum_{j:(i,j)\in L}(g^*)_{i,j}^{md}(p)+\sum_{j:(j,i)\in L}(g^*)_{j,i}^{md}(p))\\
&+\sum_{m,(i,j)\in L}(f_{i,j}^m-(f^*)_{i,j}^m)w_{i,j}^m\\
\le&\sum_{m,d,i}(p_i^{md}-(p^*)_i^{md})(\sum_{j:(i,j)\in L}(g^*)_{i,j}^{md}(p)\\
&-\sum_{j:(j,i)\in L}(g^*)_{j,i}^{md}(p)-\sum_{j:(i,j)\in L} g_{i,j}^{md}(p)\\
&+\sum_{j:(j,i)\in L} g_{j,i}^{md}(p))+\sum_{m,(i,j)\in L}(f_{i,j}^m-(f^*)_{i,j}^m)w_{i,j}^m\\
=&-\sum_{m,d,i,j}((p^*)_i^{md}-(p^*)_j^{md})((g^*)_{i,j}^{md}(p)-g_{i,j}^{md}(p))\\
&+\sum_{m,d,i,j}(p_i^{md}-p_j^{md})((g^*)_{i,j}^{md}(p)-g_{i,j}^{md}(p))\\
&+\sum_{m,(i,j)}(f_{i,j}^m-(f^*)_{i,j}^m)w_{i,j}^m,
\end{aligned}$$

where the second inequality comes from the marginal utility $U_m'(\cdot)$ being a decreasing function and the optimality conditions. Note that by relation (29), $(p_i^{md}-p_j^{md})(g^*)_{i,j}^{md} \le [p_i^{md}-p_j^{md}]^+(g^*)_{i,j}^{md} \le [p_i^{md}-p_j^{md}]^+(f^*)_{i,j}^m$ and $(p_i^{md}-p_j^{md})g_{i,j}^{md} = [p_i^{md}-p_j^{md}]^+ f_{i,j}^m$. Thus,

$$\begin{aligned}
\dot{V}(p,f) \le& -\sum_{m,d,i,j}((p^*)_i^{md}-(p^*)_j^{md})((g^*)_{i,j}^{md}(p)-g_{i,j}^{md}(p))\\
&+\sum_{m,i,j}(\sum_d[p_i^{md}-p_j^{md}]^+-w_{i,j}^m)(f^*)_{i,j}^m\\
&+(w_{i,j}^m-\sum_d[p_i^{md}-p_j^{md}]^+)f_{i,j}^m\\
=&-\sum_{m,d,i,j}((p^*)_i^{md}-(p^*)_j^{md})((g^*)_{i,j}^{md}(p)-g_{i,j}^{md}(p)).
\end{aligned}$$

Since g^* maximizes $\sum_{m,d,i,j}((p^*)_i^{md}-(p^*)_j^{md})(g^*)_{i,j}^{md}$, $\dot{V}(p,f) \le 0$. This implies that the system converges to an invariant set specified by $\dot{V}(p,f)=0$. Furthermore, from the above proof, $\dot{V}(p,f)=0$ only if p, f, g and x satisfy the optimality conditions for convex problem **P2**. ■

C. Implementation of Price Feedback

Since the scheme is destination-based, each packet need to carry a vector of destination identities in the packet header, in addition to coding vector. Each node i keeps a separate virtual queue p_i^{md} as congestion price for each multicast session m and destination $d \in D_m$. The arrival and the departure of these queues evolve as follows. When a packet is received at node i, i will check the destination vector in the header of this packet. If this packet is intended for destination d , the queue size p_i^{md} will increase by one; Otherwise, the virtual queue size will remain the same. When a packet is sent out at node i, i will check the destination vector of this packet. If this packet is intended for destination d , the queue size p_i^{md} will decrease by one; Otherwise, the virtual queue size will remain the same. Note that, here we use back-pressure to do rate control. The source nodes s adjust the sending rate according to local congestion prices at s, and the congestion in the network is propagated to the source node through back-pressure.

Remarks: There may exist several ways to solve the system problems **P1** and **P2**. The challenge is to find distributed solutions that respect as much as possible the information constraints of the Internet and can be implemented at the

sources and routers. This requires to minimize information and respect signaling mechanism for adaptive control in Internet as much as possible. So, we choose not to relax all the constraints when solving for the duals. Besides the equilibrium, dynamics are also important in our consideration. In section IV, in order to avoid "tree" oscillation, we achieve flow control through a combination of fast timescale congestion control and slow, traffic engineering timescale traffic splitting.

The two sets of decentralized controllers developed in sections IV and V can coexist: some multicast sessions adopt the algorithm with given coding subgraphs and other sessions adopt the algorithm without given coding subgraphs; and they are coupled through the flow balance equations at nodes and capacity constraints at links. Also, unicasting can be seen as special case of multicasting. Mathematically, in the system model **P1**, network coding comes into action through constraint $H_{lr}^m x_r^m \leq y_l^m$, and in system model **P2**, network coding comes into action through constraint $g_{i,j}^{md} \leq f_{i,j}^m$. It is straightforward to include uncoded unicast flows into the system models and carry out these algorithms in the same way, with only slightly more complicated notation.

VI. Numerical Examples

In this section, we provide numerical examples to complement the analysis in previous sections. We consider a simple network shown in the left graph in Figure 3. The network is assumed to be undirected and each link has equal capacities in both directions. Assume that there are two multicast sessions, session one with source node s and destinations x and y and session two with source node t and destination u and z, with the same utility $U_m(x_m) = \log(x_m)$. We have chosen such a small, simple topology to facilitate detailed discussion of the results.

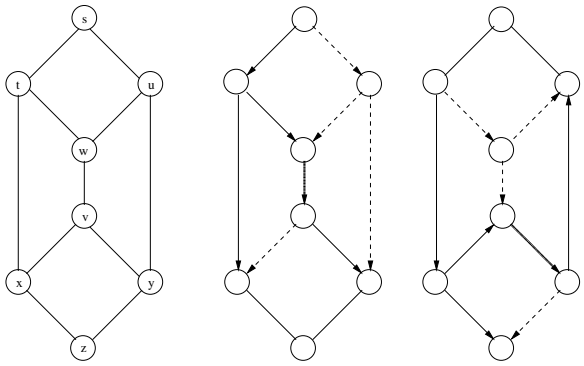


Fig. 3. A simple network with two multicast sessions. The given coding subgraphs for sessions 1 and 2 are shown in the middle and right graphs respectively. For each session, the first tree is indicated by solid arrows, the second by dashed arrows, and the overlapping segments by bold arrows.

A. Muticasting with Given Coding Subgraphs

We assume that the given coding subgraphs for sessions one and two are those shown in the middle and right graphs of Figure 3 respectively. The subgraph for each session decomposes into two multicast trees in the same way as in Figure 2. For simplicity, we assume the following link capacities: link (s,t) has 2 units of capacity, links (t,x) and (v,y) have 5 units of capacity, links (s,u), (u,w) and (y,z) have 1 unit of capacity and all other links have 3 units of capacity.

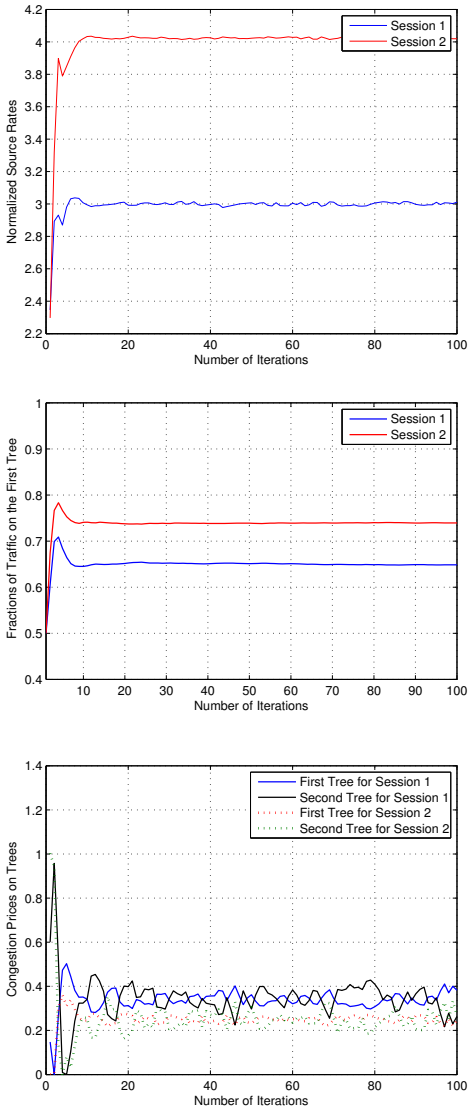


Fig. 4. The evolution of source rates (upper panel), the evolution of traffic split vectors (middle panel), and the evolution of congestion prices over different multicast trees (lower panel) versus the number of iterations of the tree adaptation algorithm with stepsize $\delta t \kappa_m = 0.01$ for the example network with given coding subgraphs.

We consider the following discrete-time implementation of flow control (6)-(9). Periodically with period δt, the source rate, session allocation and congestion price are updated according to[5]

$$x^m \longleftarrow (U_m')^{-1}\Big(\sum_r \alpha_r^m \sum_l H_{l,r}^m p_{l,r}^m\Big), \tag{31}$$

$$y_l^m \longleftarrow y_l^m + \delta t \varepsilon_l [p_l^m - \eta_l[p]]_{y_l^m}^+, \tag{32}$$

$$p_{l,r}^m \longleftarrow [p_{l,r}^m + \delta t \gamma_l (H_{lr}^m \alpha_r^m x^m(p) - y_l^m(p))]_0^+, \tag{33}$$

and periodically with period $\Delta t \gg \delta t$, the traffic split variable

[5]Care should be taken in the implementation of equations (32), (34) and (36) to guarantee the corresponding equality constraints.

is adjusted according to

$$\alpha_r^m \longleftarrow \alpha_r^m + \Delta t \kappa_m [E^m[p] - p_r^m]^+_{\alpha_r^m}. \quad (34)$$

Figure 4 shows the evolution of source rates (upper panel) versus the number of iterations of the outer loop tree adaptation algorithm and the evaluation of traffic split vectors (middle panel) with stepsize $\Delta t \kappa_m = 0.01$. It can be seen from the plots that the source rates are well within 5% of their optimal values after 5 iterations, and the traffic split vectors are well within 5% of their optimal values after 10 iterations. In this simulation, the inner loop congestion control algorithm runs 500 iterations before each run of the tree adaptation algorithm. Comparable performance is observed even if the number of inner loop iterations is as low as 100. So, the convergence of the whole rate control algorithm is very fast.

In practice, the end users can dynamically control the number of iterations, by monitoring the congestion prices over different multicast trees. The lower panel of Figure 4 shows the evolution of the congestion prices over different trees versus the number of iterations of the tree adaptation algorithm. We can, for instance, specify a threshold value and decide the whole algorithm has converged when the relative differences in price over different multicast trees are less than the threshold value. The users can also set the stepsize of the tree adaptation algorithm dynamically. When the price differences over different trees are large, the user can choose a large stepsize, and when the differences are small, he can choose a small stepsize.

B. Muticasting without Given Coding Subgraphs

We now consider the same network but without given coding subgraphs. The distributed algorithm developed in section V will go through the whole network (the undirected graph on the left side in Figure 3) to find capacitated coding subgraphs that maximize the aggregate utility. For this example, we assume the following link capacities: links (s,t), (t,x) and (x,v) have 2 units of capacity, links (t,w), (w,v) and (v,y) have 3 units of capacity and all other links have 1 unit of capacity.

We consider the following discrete-time implementation of congestion control and session allocation (27)-(30). Periodically with period δt, the source rate, session allocation and congestion price are updated according to

$$x_m \longleftarrow {U'_m}^{-1}(\sum_{d \in D_m} p_{s_m}^{md}). \quad (35)$$

$$f_{i,j}^m \longleftarrow f_{i,j}^m + \delta t \varepsilon_{i,j}[w_{i,j}^m - \eta_{i,j}[w]]^+_{f_{i,j}^m}, \quad (36)$$

$$p_i^{md} \longleftarrow [p_i^{md} + \delta t \gamma_i (x_i^{md}(p) - \sum_{j:(i,j)\in L} g_{i,j}^{md}(p) + \sum_{j:(j,i)\in L} g_{j,i}^{md}(p))]_0^+. \quad (37)$$

Figure 5 shows the evolution of the source rates with stepsize $\delta t \varepsilon_{i,j} = \delta t \gamma_i = 0.01$. We see that the source rates approach the corresponding optimal quickly. The simulation result also shows coding occurs over the same subgraphs as those in Fig.3: 2 units of traffic of session one is coded over link (w,v) and 2 units of traffic of session two is coded over link (v,y). It is not difficult to check that those are optimal source rates and coding subgraphs. In order to study the impact of different choices of the stepsize on the convergence of the algorithm, we have run simulations with different stepsizes. We found that the smaller the stepsize, the slower the convergence and the closer to the optimal, which is a general characteristic of any gradient based method. So, there is a tradeoff between convergence speed and optimality. In practice, the end user can first choose large stepsizes to ensure fast convergence, and subsequently, the stepsizes can be reduced once the source rate starts oscillating around some mean value.

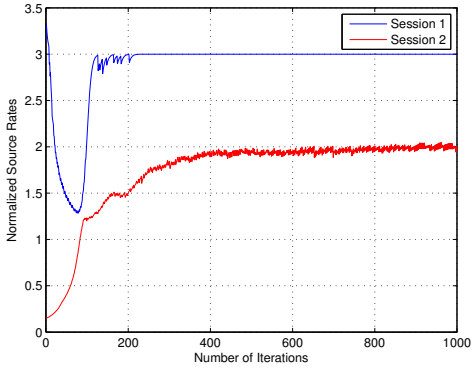


Fig. 5. The evolution of source rates with step size $\delta t \varepsilon_{i,j} = \delta t \gamma_i = 0.01$ for the example network without given coding subgraph.

C. Comparison of the Two Algorithms

To compare the performance of the two rate control algorithms, we consider the same network, with 1 unit capacity for each link. Figure 6 shows the evolution of the source rates versus the number of iterations of the tree adaptation algorithm for the case with given coding subgraphs as shown in the right side graph of Figure 3, and the evolution of the source rates for the case without given coding subgraphs. We see that the throughput achieved for the case without given subgraphs is larger than that for the case with given coding subgraphs. This is expected, since the capacity region for the case with given coding subgraph is a subset of the capacity region with the coding subgraphs unspecified.

VII. Extension to Wireless Networks

In the previous sections, we have considered congestion control for multicast with network coding in wired networks. Here, we briefly discuss its extension to wireless networks.

The wireless case is much more complicated. On the one hand, wireless channel is a shared medium and interference limited. We need to avoid simultaneous interfering transmissions. On the other hand, we want to exploit wireless multicast advantage – a single node's transmission can be received by multiple neighboring nodes, in order to achieve efficient channel utilization. We represent wireless transmissions by

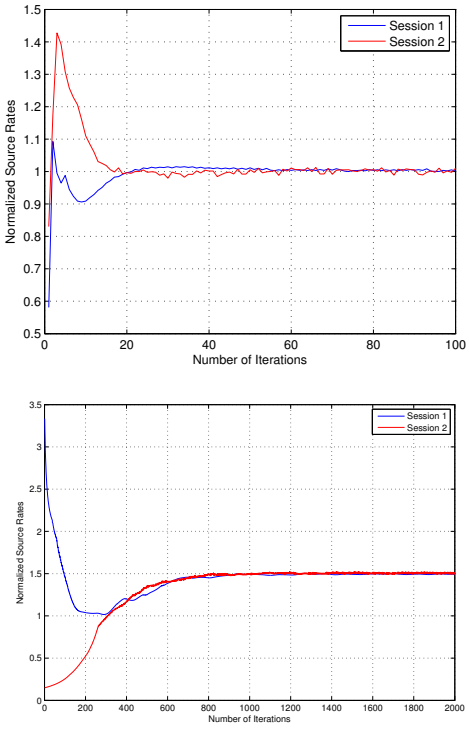


Fig. 6. The evolution of source rates for the case with given coding subgraph (upper panel) and for the case without given coding subgraph (lower panel).

hyperarcs,[6] denoted by (i, N_i), where i is the transmitting node and $N_i = \{j|(i,j) \in L, j \in N\}$ is the set of i's receiving neighbors. We assume a static topology and each hyperarc (i, N_i) has a finite fixed broadcast capacity c_{i,N_i} packets per second when active. Denote by Π the capacity region at the link layer. By the standard time sharing argument, Π is a convex hull of these capacity vectors of all interference-free schedules of hyperarcs. Given a hyperarc flow vector h, the schedulability constraint says that h should satisfy $h \in \Pi$.

Network coding allows flows for different destinations of a multicast session within a hyperarc to share capacity by being coded together, and the physical flows of different multicast sessions within a hyperarc should share the hyperarc capacity. These constraints can be expressed as

$$\sum_{j \in N_i} g_{i,j}^{md} \leq f_{i,N_i}^m, \ \forall d \in D_m, \tag{38}$$

$$\{\sum_m f_{i,N_i}^m\} \in \Pi, \tag{39}$$

where again $g_{i,j}^{md}$ is the information flow for destination d of multicast session m we defined in section III, f_{i,N_i}^m gives the the amount of broadcast capacity of hyperarc (i, N_i) that is allocated to session m, and the schedulability constraint (39) simply says that the aggregate capacity within the hyperarchs should be in the feasible capacity region.

[6]Hyperarc is a rather general construction, see, e.g., [11]. It can correspond to a single transmission from node i to any subset of its neighboring nodes. Here, for simplicity of presentation, we assume that each transmitting node is associated with only one hyperarc (i, N_i). The extension to the situation with general hyperarcs is straightforward.

We will focus on congestion control for multicast in wireless networks without given coding subgraphs. The case with given coding subgraphs can be handled in a similar way. With the above constraints, we formulate network resource allocation as the following utility maximization problem

$$\begin{aligned} \mathbf{PW}: \quad \max_{x,g,f} \quad & \sum_m U_m(x^m) \\ \text{subject to} \quad & \sum_{j:(i,j)\in L} g_{i,j}^{md} - \sum_{j:(j,i)\in L} g_{j,i}^{md} = x_i^m, i \neq d, \forall d, m \\ & \sum_{j \in N_i} g_{i,j}^{md} \leq f_{i,N_i}^m, \ \forall d, m \\ & \{\sum_m f_{i,N_i}^m\} \in \Pi, \end{aligned}$$

where again $x_i^m = x^m$ if $i = s_m$ and $x_i^m = 0$ otherwise.

Problem **PW** has similar structure to problem **P2**, so we can solve it similarly. However, in order to integrate with wireless scheduling, we will propose a different congestion controller, with the session allocation component replaced by a scheduling component, and present them in discrete-time.

Consider the Lagrangian of the system problem **PW** with respect to the flow balance constraints,

$$\begin{aligned} L(p,x,g,f) = & \sum_m U_m(x^m) - \sum_{i,m,d\in D_m} p_i^{md}(x_i^m \\ & - \sum_{j:(i,j)\in L} g_{i,j}^{md} + \sum_{j:(j,i)\in L} g_{j,i}^{md}). \end{aligned}$$

Following similar procedures as in section V, we can obtain the following discrete-time congestion control and scheduling algorithm.

Congestion control: At time t, each source node s_m adjusts its sending rate according to local congestion price at the source node,

$$x_m(t) = {U'_m}^{-1}(\sum_{d\in D_m} p_{s_m}^{md}(t)). \tag{40}$$

Note that

$$\begin{aligned} & \max_{g,f} \sum_{i,m,d} p_i^{md}(\sum_j g_{i,j}^{md} - \sum_j g_{j,i}^{md}) \ \ s.t. \ \sum_{j\in N_i} g_{i,j}^{md} \leq f_{i,N_i}^m \\ = & \max_{g,f} \sum_{i,j,m,d} g_{i,j}^{md}(p_i^{md} - p_j^{md}) \ \ s.t. \ \sum_{j\in N_i} g_{i,j}^{md} \leq f_{i,N_i}^m \\ = & \max_f \sum_{i,m,d} f_{i,N_i}^m \max_j [p_i^{md} - p_j^{md}]^+, \end{aligned}$$

and further,

$$\begin{aligned} & \max_f \sum_{i,m,d} f_{i,N_i}^m \max_j [p_i^{md} - p_j^{md}]^+ \ s.t. \ \{\sum_m f_{i,N_i}^m\} \in \Pi \\ = & \max_f \sum_{i,m} f_{i,N_i}^m \sum_d \max_j [p_i^{md} - p_j^{md}]^+ \ s.t. \ \{\sum_m f_{i,N_i}^m\} \in \Pi \\ = & \max_f \sum_i f_{i,N_i} \max_m \sum_d \max_j [p_i^{md} - p_j^{md}]^+ \ s.t. \ \{f_{i,N_i}\} \in \Pi. \end{aligned}$$

Each node i collects congestion price information from its neighbor j, find multicast session $m_{i,N_i}(t)$ such that

$$m_{i,N_i}(t) = \arg\max_m \sum_{d\in D_m} \max_{j\in N_i} [p_i^{md}(t) - p_j^{md}(t)]^+, \tag{41}$$

and calculates the differential price

$$w_{i,N_i}(t) = \sum_d \max_{j \in N_i} [p_i^{m_{i,N_i}(t)d}(t) - p_j^{m_{i,N_i}(t)d}(t)]^+.$$

Scheduling: Over hyperarc (i, N_i), a random linear combination of data of multicast session m_{i,N_i} to all destinations d such that $\max_{j \in N_i} p_i^{m_{i,N_i}d}(t) - p_j^{m_{i,N_i}d}(t) > 0$ is sent at rate $\tilde{f}_{i,N_i}$, where $\{\tilde{f}_{i,N_i}\}$ is an extreme point maximizer to the following hyperarc scheduling problem:

$$\max_f \sum_i f_{i,N_i} w_{i,N_i}(t) \ s.t. \ \{f_{i,N_i}\} \in \Pi. \tag{42}$$

Mathematically, this is equivalent to solving the primal variable g by the following assignment

$$g_{i,j}^{md}(t) = \begin{cases} \tilde{f}_{i,N_i} & \text{if } m = m_{i,N_i}(t) \\ & \quad \& \ j = \arg\max_{k \in N_i} [p_i^{md}(t) - p_k^{md}(t)]^+ \\ & \quad \& \ [p_i^{md}(t) - p_j^{md}(t)]^+ > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{43}$$

Congestion price update: Each node i updates its price with respect to multicast session m and destination $d \in D_m$, according to

$$\begin{aligned} p_i^{md}(t+1) &= [p_i^{md}(t) + \gamma_t(\ x_i^{md}(p(t)) \\ &- \sum_{j:(i,j)\in L} g_{i,j}^{md}(p(t)) + \sum_{j:(j,i)\in L} g_{j,i}^{md}(p(t)) \ )]^+, \end{aligned} \tag{44}$$

and passes the price p_i^{md} to all its neighbors.

We see that the above congestion control, network coding and scheduling algorithm is similar to the distributed algorithm (35)-(37). While for wired networks we do not consider the details of session scheduling at a link, for wireless networks we explicitly study session scheduling. In addition to session scheduling (41) within a hyperarch, there is also a hyperarc scheduling component (42). The hyperarc scheduling (42) is centralized, and is NP-complete in general. We are studying distributed approximation algorithms to solve (42), which will be reported elsewhere.

The algorithm (40)-(44) is a subgradient algorithm. We can straightforwardly apply either the standard convergence results for the subgradient method [25] or the convergence analysis in, e.g,, [22], [3] to establish its convergence. We will not elaborate on this.

VIII. Conclusions

We have presented two models for congestion control for multicast flows with network coding, one for networks with given coding subgraphs, and one where such subgraphs are found dynamically. Correspondingly, we developed two sets of decentralized controllers for congestion control. With random network coding, both sets of controllers can be implemented in a distributed manner, and work at transport layer to adjust source rates and at network layer to carry out network coding. We prove that the proposed controllers converge to the global optimum for each model. Numerical examples are provided to complement our theoretical analysis. We will further study the practical implementation of our congestion controllers. We are also studying their stability under propagation delay. Also, how to obtain optimal coding subgraphs based on general cost criteria is an interesting problem. Solving this problem will further facilitate the practical deployment of network coding in real networks.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, Network information flow, *IEEE Trans. on Information Theory*, 46:1204-1216, 2000.

[2] D. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena scientific, 1999.

[3] L. Chen, S. Low, M. Chiang and J. Doyle, Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks, *Proc. IEEE Infocom*, 2006.

[4] L. Chen, T. Ho, M. Chiang, S. H. Low, and J. Doyle, Optimization Based Rate Control for Multi-cast with Network Coding, *Proc. IEEE Infocom*, 2007.

[5] P. A. Chou, Y. Wu and K. Jain, Practical network coding, *Proc. Allerton Conference on Communication, Control and Computing*, 2003.

[6] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi and M. Effros, Capacity of wireless erasure networks, *IEEE Trans. on Information Theory*, 2006.

[7] S. Deb and R. Srikant, Congestion control for fair resource allocation in networks with multicast flows, *Proc. CDC*, 2001.

[8] R. Gallager, A minimum delay routing algorithm using distributed computation, *IEEE Transactions on Communication*, 25(1):73-85, 1977.

[9] T. Ho, R. Koetter, M. Medard, D. R. Karger and M. Effros, The Benefits of Coding over Routing in a Randomized Setting, *Proc. of IEEE International Symposium on Information Theory*, 2003.

[10] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, A Random Linear Network Coding Approach to Multicast, *IEEE Transactions on Information Theory*, 52(10):4413–4430, October 2004.

[11] T. Ho and H. Viswanathan, Dynamic algorithms for multicast with intra-session network coding, *Proc. Allerton Conference on Communication, Control and Computing*, 2005.

[12] K. Kar, S. Sarkar and L. Tassiulas, Optimization based rate control for multirate multicast sessions, *Proc. IEEE Infocom*, 2001.

[13] F. P. Kelly, A. K. Maulloo and D. K. H. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, *Journal of Operations Research Society*, 49(3):237-252, March 1998.

[14] H.K. Khalil, *Nonlinear Systems*, Prentice-Hall, 1996.

[15] R. Koetter, M. Medard, An Algebraic Approach to Network Coding, *IEEE/ACM Transactions on Networking*, 11:782-795, 2003.

[16] S. Kunniyur and R. Srikant, End-to-end congestion control schemes: Utility functions, random losses and ECN marks, *IEEE/ACM Transactions on networking*, 11(5):689-702, October 2003.

[17] S.-Y. R. Li, R. W. Yeung, and N. Cai, Linear network coding, *IEEE Transactions on Information Theory*, 49:371-381, 2003.

[18] S. H. Low and D. E. Lapsley, Optimal flow control, I: Basic algorithm and convergence, *IEEE/ACM Trans. on networking*, 7(6):861-874, 1999.

[19] D. S. Lun, M. Medard and M. Effros, On coding for reliable communication over packet networks, *Proc. of Allerton Conference on Communication, Control, and Computing*, 2004.

[20] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho and E. Ahmed, Minimum-cost multicast over coded packet networks, *IEEE Trans. Inform. Theory*, 2006.

[21] M. Neely, E. Modiano and C. Rohrs, Dynamic power allocation and routing for time varying wireless networks *Proc. IEEE Infocom*, 2003. Journal version, *IEEE J. Sel. Area Comm.*, 23(1):89-103, 2005.

[22] M. Neely, E. Modiano and C. Li, Fairness and optimal stochastic control for heterogeneous networks, *Proc. IEEE Infocom*, 2005.

[23] F. Pagnini, Congestion control with adaptive multipath routing based on optimization, *Proc. CISS*, 2006.

[24] Saswati Sarkar and Leandros Tassiulas. A framework for routing and congestion control for multicast information flows. *IEEE Transactions on Information Theory*, 2002.

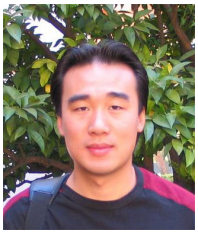
[25] N. Z. Shor, *Monimization Methods for Non-Differentiable Functions*, Springer-Verlag, 1985.

[26] L. Tassiulas and A. F. Ephremides, Stability Properties of Constrained Queuing Systems and Scheduling Policies for Maximum Throughput in Multihop Networks, *IEEE Transactions on Information Theory*, 1992.

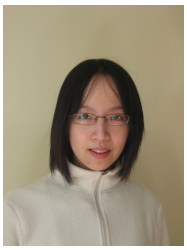
[27] J. G. Wardrop, Some theoretical aspects of road traffic research, Proceedings, Institute of Civil Engineers, PART II, Vol.1, pp. 325-378.

[28] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu and S. Y. Kung, Network Planning in Wireless Ad Hoc Networks: A Cross-Layer Approach, *IEEE Journal on Selected Areas in Communications*, 2005.

[29] Y. Wu, M. Chiang and S.Y. Kung, Distributed utility maximization for network coding based multicasting: A critical cut approach, *Proc. IEEE NetCod*, 2006.
[30] Y. Wu and S.Y. Kung, Distributed utility maximization for network coding based multicasting: A shortest path approach, *IEEE Journal on Selected Areas in Communications*, 2006.
[31] Y. Xi and E. M. Yeh, Distributed algorithms for minimum cost multicast with network coding, *Proc. Allerton Conference*, 2005.
[32] R. W. Yeung, Multilevel diversity coding with distortion, *IEEE Trans. on Information Theory*, 1995.



Lijun Chen (M'05) is an Assistant Professor in Telecommunications at University of Colorado at Boulder. He received a B.S. from University of Science and Technology of China, M.S. from Institute of Theoretical Physics, Chinese Academy of Sciences and from University of Maryland at College Park, and Ph.D. from California Institute of Technology. He was a co-recipient of the Best Paper Award at the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS) in 2007. His current research interests are in communication networks, smart grids, optimization, game theory and their engineering application.



Tracey Ho (M'06) is an Assistant Professor in Electrical Engineering and Computer Science at the California Institute of Technology. She received a Ph.D. (2004) and B.S. and M.Eng degrees (1999) in Electrical Engineering and Computer Science (EECS) from the Massachusetts Institute of Technology (MIT). She was a co-recipient of the 2009 Communications & Information Theory Society Joint Paper Award. Her primary research interests are in information theory, network coding and communication networks.

Mung Chiang (F'12) is a Professor of Electrical Engineering at Princeton University, and an affiliated faculty in Applied and Computational Mathematics, and in Computer Science. He received his B.S. (Hons.), M.S., and Ph.D. degrees from Stanford University in 1999, 2000, and 2003, respectively, and was an Assistant Professor 2003-2008 and an Associate Professor 2008-2011 at Princeton University. His research on networking received the IEEE Kiyo Tomiyasu Award (2012), a U.S. Presidential Early Career Award for Scientists and Engineers (2008), several young investigator awards, and a few paper awards including the IEEE INFOCOM Best Paper Award (2012). His inventions resulted in a few technology transfers to commercial adoption, and he received a Technology Review TR35 Award (2007) and founded the Princeton EDGE Lab in 2009. He served as an IEEE Communications Society Distinguished Lecturer in 2012-2013, and wrote an undergraduate textbook: "Networked Life: 20 Questions and Answers."

Steven H. Low (F'08) is a Professor of the Computing & Mathematical Sciences and Electrical Engineering Departments at Caltech, and an adjunct professor of both Swinburne University, Australia and Shanghai Jiao Tong University, China. Before that, he was with AT&T Bell Laboratories, Murray Hill, NJ, and the University of Melbourne, Australia. He was a co-recipient of IEEE best paper awards, the R&D 100 Award, and an Okawa Foundation Research Grant. He was on the editorial boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Automatic Control, ACM Computing Surveys, Computer Networks Journal, NOW Foundations and Trends in Networking. He is a Senior Editor of the IEEE Journal on Selected Areas in Communications. He received his B.S. from Cornell and PhD from Berkeley, both in EE.

John C. Doyle (BS/MS, EE, MIT, 1977; PhD, Math, UC Berkeley, 1984) is John G Braun Professor of Control and Dynamical Systems, EE and BioE at Caltech. Current research is in theoretical foundations, for complex networks in engineering and biology, focusing on architecture, and for multiscale physics. Early work was in the mathematics of robust control, including recent extensions to nonlinear and hybrid systems. His group has contributed to many software projects, including the Robust Control Toolbox (muTools), SOSTOOLS, SBML (Systems Biology Markup Language), and FAST (Fast AQM, Scalable TCP). Prizes include the IEEE Baker and Trans Aut Control Axelby (twice), and ACM Sigcomm and AACC American Control Conferences best papers. Individual awards include the AACC Eckman and the IEEE Control Systems Field and Centennial Outstanding Young Engineer Awards. He has held national and world records and championships in various sports.