Optimization Based Rate Control for Communication Networks with Inter-session Network Coding

Abdallah Khreishah
School of ECE
Purdue University
West Lafayette, IN 47906
Email: akhreish@purdue.edu

Chih-Chun Wang
School of ECE
Purdue University
West Lafayette, IN 47906
Email: chihw@purdue.edu

Ness B. Shroff
Departments of ECE and CSE
The Ohio State University
Columbus, OH 43210, USA
Email:shroff@ece.osu.edu

***Abstract*—In this paper we develop a distributed rate control algorithm for multiple-unicast-sessions when network coding is allowed. Building on our recent flow-based characterization of network coding, we formulate the problem as a convex optimization problem. The formulation exploits pairwise coding possibilities between any pair of sessions, where the objective function is the sum of the utilities based on the rates supported by each session. With some manipulation on the Lagrangian of the formulated problem, a distributed algorithm is developed with no interaction between intermediate nodes, and each source having the freedom to choose its own utility function. The only information required by the source is the weighted sum of the queue length updates of each link, which can be piggy-backed on the acknowledgment messages. In addition to the optimal rate control algorithm, we propose a decentralized Pairwise Random Coding Scheme (PRC) that is optimal when a sufficiently large finite field is used for network coding. The convergence of the rate control algorithm is proved analytically and verified by extensive simulations. Simulations also demonstrate the advantage of our algorithm over the state-of-the-art in terms of throughput and fairness.**

***Index Terms*—Inter-session Network Coding, Multiple-unicast-sessions Problem, Rate Control, Distributed Algorithm, Capacity Region, Coding Scheme, Fairness.**

I. Introduction and related work

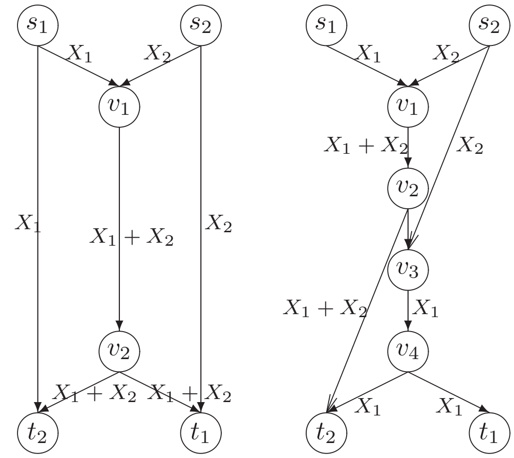
Over the last several years, there has been tremendous interest in the study of network optimization techniques to maximize network performance, while at the same time achieving fairness across flows (see, for example, [3], [7], [13], [15], [22], [23], [25]).

More recently, a new area of research has emerged called network coding that has the potential to further increase the achievable throughput by packet mixing at intermediate nodes [1]. While network coding has primarily been used to improve network capacity, we believe that it can also be used to provide fairness. A key challenge is to provide a solution that not only satisfies optimality considerations but is distributed and easy to implement. To that end, in this paper, we use inter-session network coding, where coding is permitted between different sessions, to develop a *distributed algorithm* that maximizes network performance and enhances fairness. Our focus will be on the study of multiple unicast sessions.

To see that network coding can be used to achieve fairness, consider the following classical butterfly topology, shown in Figure 1(a). As is the convention in the network coding literature, we use the term "routing" to describe the solution for which no packet mixing is used. In this simple butterfly configuration, we assume that each link can sustain a throughput of at most 1 packet per second (in subsequent discussions we drop the units). Senders s_1 and s_2 want to send packets to receivers t_1 and t_2, respectively. For this topology, if only routing solutions are permitted, we could achieve strict fairness (i.e., transmissions at equal rates from both sources) with the rate for each sender being no higher than 0.5. However, using network coding, it can be easily seen that we can sustain unit rates for both senders simultaneously. Another example is the so-called grail topology shown in Figure 1(b) introduced in our previous work [28], where each link has a unit capacity and the sending and receiving requirements are the same as in the butterfly example. Here, the maximum of the sum rate of the two senders using routing is 2. However, if we operate the network at that point which is achieved by s_2 taking full advantage of the network while s_1 remaining silent, we will have a clearly undesirable and unfair situation. Again, network coding can help resolve the problem, as one can achieve a total rate of 2 while being strictly fair and both sources can maintain unit rates simultaneously. For both topologies, the network coding solution on each link that achieves the strictly fair rates is illustrated in Figure 1.

In the literature, beginning with the seminal paper [1], network coding for a single multicast session has been extensively studied. Follow-up works include [20], which shows that linear coding is sufficient for multicast. In [17], the authors develop a useful algebraic approach to network coding and [11] provides a distributed implementation of network-coding-based multicast using random network coding. Several other interesting works can be found in [12], [21], [24], [30].

This work has been sponsored in part by NSF grants CNS 0626703 and CCF 0635202



(a) The butterfly. (b) The grail.

Fig. 1. Two examples of graphs benefited from network coding and their corresponding network coding schemes.

Network coding for multiple unicast/multicast sessions has been less studied. In [8] it was shown that linear coding is no longer capable of achieving the optimal capacity region in the multiple unicast/multicast case in contrast with the single multicast session. Since then, many studies have targeted suboptimal solutions to the multi-session unicast problem using network coding. For wireless networks, the nature of links further enriches coding possibility, as broadcasting can be achieved without power penalty. Therefore, a much smaller number of transmissions is necessary than its wired counterpart. Based on the broadcasting nature of wireless networks, opportunistic exclusive OR (XOR) coding is introduced in [14], and further improved and analyzed in [26].

For wired networks, which are most related to our paper, the TRLKM capacity region was introduced in [27], which captures any possible butterfly structure in the network. Two distributed algorithms for the TRLKM region using back-pressure techniques were provided in [9] and [10]. In these two algorithms, exchanging queue length information between intermediate nodes is required. The purpose of the exchange is to determine the location of the encoding, decoding, and remedy generating nodes in [9], and for remedy packets requests in [10]. Like any back pressure algorithm, these two algorithms focus on stabilize the given network traffic instead of dynamic rate control for fairness improvement.

In this paper, our main contributions are as follows:

1) The development of a distributed algorithm with rate control and utility maximization for *inter-session* network coding for multiple unicast flows. Existing works either focus on rate control for the routing solutions [13], [15], [22], [23], [25], or rate control that only takes advantages of *intra-session network coding* [5], [24], [29], [31]. Our result is the first rate control and utility optimization scheme that takes advantages of the *inter-session network coding*.
2) Our result is developed based on finding good paths rather than finding specific structures in the network (such as the butterfly structures in [27]). This enables more efficient solutions since one can leverage upon existing work on how to choose good paths through the network. Further, we show empirically that the capacity region obtained via our approach can be considerably larger than obtained via the pattern search algorithm [9], [10], [27].
3) The results we obtain are not restricted to XOR based coding. Our proposed Pairwise Random Coding (PRC) scheme, which is a modified version of the random linear coding scheme in [11], facilitates developing a fully distributed algorithm. Combining the distributed rate control and the decentralized coding scheme, we eliminate information exchange between intermediate nodes, previously used in [9] and [10].

The rest of the paper is organized as follows. In Section II, our previous results in [28] are briefly reviewed to establish the corresponding theoretical background. In Section III, we describe the system settings and the formulation of our optimization problem. In Section IV, we solve the dual problem to obtain the distributed rate control algorithm. The PRC coding scheme is introduced in Section V followed by a discussion on the practical implementation issues in Section VI. Section VII is devoted to simulation results. We conclude the paper in Section VIII.

II. Preliminary results

In our previous work [28], we have studied the problem of network coding with two simple unicast sessions for general directed acyclic graphs (DAG). Let $Q_{v,w}, P_{v,w}$ denote paths from node v to node w. $\mathcal{P}$ is a set of paths, and the number of coinciding paths at link n, $\mathsf{ncp}_{\mathcal{P}}(n)$ is the number of paths in the set $\mathcal{P}$ that use link n. E denotes the set of all links in the network. The main results in [28] are:

Theorem 1: For a DAG with unit capacity links and two co-existing unicast sessions between the source-sink pairs (s_1,t_1), (s_2,t_2), a linear network coding scheme that can support unit rates for both sessions exists if and only if one of the following two conditions holds.

- [Condition 1] There exists a collection $\mathcal{P}$ of two paths P_{s_1,t_1} and P_{s_2,t_2}, such that $\max_{n\in E} \mathsf{ncp}_{\mathcal{P}}(n) \leq 1$.
- [Condition 2] There exist a collection $\mathcal{P}$ of three paths $\{P_{s_1,t_1}, P_{s_2,t_2}, P_{s_2,t_1}\}$, and a collection $\mathcal{Q}$ of three paths $\{Q_{s_1,t_1}, Q_{s_2,t_2}, Q_{s_1,t_2}\}$, such that $\max_{n\in E} \mathsf{ncp}_{\mathcal{P}}(n) \leq 2$ and $\max_{n\in E} \mathsf{ncp}_{\mathcal{Q}}(n) \leq 2$.

If Condition 1 is satisfied, the problem is feasible with routing only. The butterfly network in Figure 1(a) satisfies only Condition 2 with $P_{s_1,t_1} = \{s_1, v_1, v_2, t_1\}, P_{s_2,t_2} = \{s_2, v_1, v_2, t_2\}, P_{s_2,t_1} = \{s_2, t_1\}, Q_{s_1,t_1} = \{s_1, v_1, v_2, t_1\}, Q_{s_2,t_2} = \{s_2, v_1, v_2, t_2\}$, and $Q_{s_1,t_2} = \{s_1, t_2\}$. This means that without coding the problem is infeasible. Another example where only Condition 2 is satisfied, is the grail network in Figure 1(b) with $P_{s_1,t_1} = \{s_1, v_1, v_2, v_3, v_4, t_1\}, P_{s_2,t_2} =$

$\{s_2, v_1, v_2, t_2\}, P_{s_2,t_1} = \{s_2, v_3, v_4, t_1\}, Q_{s_1,t_1} = \{s_1, v_1, v_2, v_3, v_4, t_1\}, Q_{s_2,t_2} = \{s_2, v_3, v_4, t_2\}$, and $Q_{s_1,t_2} = \{s_1, v_1, v_2, t_2\}$.

Corollary 4 in [28]:
Consider I source-&-sink pairs and each source s_i would like to transmit R_i symbols per unit time to the corresponding sink t_i over a finite DAG. Each link n is capable of transmitting C_n symbols per time-frame with no transmission delay. The rate vector $(R_1, \cdots, R_I)$ is feasible if the original graph G can be viewed as the superposition of one graph G_r and many graphs G_p's such that (i) routing is performed for every (s_i, t_i) pair in G_r, (ii) pairwise linear network coding across (s_i, t_i) and (s_j, t_j) is performed in each G_p individually with a rate g_p, and (iii) the transmission rates $(R_1, \cdots, R_I)$ can be supported. Here $R_i = x_i + \sum_{G_p \in L(i)} g_p$, where x_i is the rate supported for session i in G_r, and $L(i)$ is the set of all G_p where pairwise network coding is performed between session (s_i,t_i) and any other session. The necessary and sufficient condition of each G_p is as stated in Theorem 1 with the modification that the capacity of all links is g_p instead of the unit rate. Each G_p is one Pairwise Intersession Coding Configuration (PICC) for two different sessions (s_i, t_i) and (s_j, t_j).

This Corollary illustrates how to use the result in Theorem 1 in a network with more than two sessions by characterizing the intersession coding opportunities as the collection of different PICCs. It describes a capacity region that we will call by the initials of the names of the authors the ($\mathcal{WS}$) region. In the next section we will describe the $\mathcal{WS}$ region by a set of constraints.

III. System model and problem formulation

We consider a finite DAG $G = (V, E)$, where V and E are the sets of all nodes and links, respectively. We denote the capacity of each link n by C_n. The problem is defined by the set of tuples $(s_i, t_i, U_i(R_i))$ $i \in 1, 2, \ldots, I$. Session i is represented by the tuple with index i, where s_i and t_i are the source and sink nodes, respectively. Packets transmitted through session i are independent of those transmitted through session j $\forall i \neq j$. U_i is a concave increasing utility function for session i that depends on the transmission rate R_i supported between s_i and t_i.

Since in the $\mathcal{WS}$ region, the rate R_i is expressed as the sum of rates with/without inter-session network coding, two sets of parameters and variables will be used in our formulation. Some parameters and variables are for the routing-only graph G_r and the others capture the inter-session network coding performed on graphs G_p in the optimization problem. For G_r, we define the parameters $J(i)$ and $\mathcal{E}_i^{nk}$, and the variable x_i^k. Let $J(i)$ represent the number of paths between s_i and t_i. If link n is used by the k-th path between s_i and t_i, where k ranges from 1 to $J(i)$, then $\mathcal{E}_i^{nk} = 1$. Otherwise it is set to zero. We define x_i^k to represent the uncoded, routing rate supported through the k-th path between s_i and t_i in G_r. $\overrightarrow{x}$ is a column vector containing x_i^k $\forall i, k$. For G_p, we define the parameters $R(i,j)$, $J(i,j)$, and H_{ij}^{nl}, and the variable g_{ij}^{lm}. $R(i,j)$ is the set of all tuples containing all possible choices of paths $\{P_{s_i,t_i}, P_{s_j,t_j}, P_{s_j,t_i}\}$ and $J(i,j) = |R(i,j)|$. Based on $R(i,j)$, H_{ij}^{nl} is defined in the following manner:

$$H_{ij}^{nl} = \begin{cases} 0 & \text{if no path in the } l\text{-th tuple in } R(i,j) \\ & \text{uses link } n \\ 1 & \text{if 1 or 2 paths in the } l\text{-th tuple in} \\ & R(i,j) \text{ use link } n \\ 2 & \text{if 3 paths in the } l\text{-th tuple in } R(i,j) \\ & \text{use link } n \end{cases}$$

In this paper we denote any possible choice of paths that can form $\mathcal{P}$ and $\mathcal{Q}$ in Theorem 1 as a *Pairwise Intersession Coding Configuration* (PICC). Any PICC between sessions i and j can be indexed by l and m jointly, namely, when the l-th tuple in $R(i,j)$ and the m-th tuple in $R(j,i)$ are used as the corresponding $\mathcal{P}$ and $\mathcal{Q}$. We denote the PICC formed by the paths in the l-th tuple in $R(i,j)$ and the paths in the m-th tuple in $R(j,i)$ by G_{ij}^{lm}, and the rate supported by pairwise network coding for i and j over G_{ij}^{lm} is denoted by g_{ij}^{lm}. We also define $\overrightarrow{g}$ as a column vector containing $g_{ij}^{lm} \forall i, j, l, m$. Therefore, the total supported rate becomes $R_i = \sum_{k=1}^{J(i)} x_i^k + \sum_{j \neq i} \sum_{l=1}^{J(i,j)} \sum_{m=1}^{J(j,i)} g_{ij}^{lm}$.

Consider a specific link n. The capacity consumed by pure routing traffic is: $\sum_{i=1}^{I} \sum_{k=1}^{J(i)} \mathcal{E}_i^{nk} x_i^k$. For any PICC represented by G_{ij}^{lm} indexed by l and m, the capacity consumed by $\mathcal{P}$ is $H_{ij}^{nl} g_{ij}^{lm}$. This is because by Theorem 1, the successful pairwise network coding requires that $\mathsf{ncp}_{\mathcal{P}}(n) \leq 2$. If all three paths in $\mathcal{P}$ use link n, then the traffic along these three paths must use two parallel links instead of a single one. Otherwise, the link share number will be three, which violates the necessary condition for pairwise network coding. The same argument holds for the traffic along the paths in $\mathcal{Q}$, the m-th tuple in $R(j,i)$, for which paths consume $H_{ji}^{nm} g_{ij}^{lm}$. From the above reasoning, the total capacity consumed by inter-session coding for G_{ij}^{lm} is the maximum of the two which is formally expressed as $\max(H_{ij}^{nl}, H_{ji}^{nm}) g_{ij}^{lm}$. Summing over all pairs of sessions $i \neq j$, and all l-th and m-th tuples of $R(i,j)$ and $R(j,i)$, the total capacity consumed by inter-session network coding becomes $\sum_{i=1}^{I} \sum_{i<j} \sum_{l=1}^{J(i,j)} \sum_{m=1}^{J(j,i)} \max(H_{ij}^{nl}, H_{ji}^{nm}) g_{ij}^{lm}$.

Let $\mathcal{N}(i,j) = \{1, \ldots, J(i,j)\}$, and $\mathcal{N}(i) = \{1, \ldots, J(i)\}$, from the above discussion, the following constraints represent $\{\overrightarrow{x}, \overrightarrow{g}\} \in \mathcal{WS}$.

$$\sum_{\substack{i \\ k \in \mathcal{N}(i)}} \mathcal{E}_i^{nk} x_i^k + \sum_{\substack{i, i<j \\ l \in \mathcal{N}(i,j) \\ m \in \mathcal{N}(j,i)}} \max(H_{ij}^{nl}, H_{ji}^{nm}) g_{ij}^{lm} \leq C_n \quad \forall n \in \{1, \ldots, |E|\}$$

$$g_{ij}^{lm} = g_{ji}^{ml} \quad \forall i < j, l, m. \tag{1}$$

Thus, our optimization problem becomes:

$$\max_{\{\overrightarrow{x},\overrightarrow{g}\}\geq 0}\sum_{i=1}^{I}U_i(\sum_{k\in\mathcal{N}(i)}x_i^k+\sum_{\substack{i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}g_{ij}^{lm}) \tag{2}$$

subject to:

$$\{\overrightarrow{x},\overrightarrow{g}\}\in\mathcal{WS}$$

IV. The rate control algorithm

By change of variable indices i and j we have

$$\sum_{\substack{i,i<j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}\max(H_{ij}^{nl},H_{ji}^{nm})g_{ij}^{lm} = \sum_{\substack{i,j<i\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}\max(H_{ij}^{nl},H_{ji}^{nm})g_{ji}^{ml}.$$

We denote $\frac{1}{2}\max(H_{ij}^{nl},H_{ji}^{nm})$ by F_{ij}^{nlm}. Since $g_{ij}^{lm}=g_{ji}^{ml}$, the constraints in (1) can be rewritten as:

$$\sum_{\substack{i\\ k\in\mathcal{N}(i)}}\mathcal{E}_i^{nk}x_i^k+\sum_{\substack{i,i<j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}F_{ij}^{nlm}g_{ij}^{lm} + \sum_{\substack{i,j<i\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}F_{ij}^{nlm}g_{ij}^{lm}\leq C_n \quad \forall n\in\{1,\ldots,|E|\}$$

$$g_{ij}^{lm}=g_{ji}^{ml}\quad \forall i<j,l,m$$

Note that even if every utility function $U_i(\cdot)$ is strictly concave, the objective function in (2) may not be strictly concave due to the presence of the linear terms $\sum_{k\in\mathcal{N}(i)}x_i^k+\sum_{i\neq j}\sum_{l=1}^{J(i,j)}\sum_{m=1}^{J(j,i)}g_{ij}^{lm}$. Thus, a direct application of standard convex optimization techniques might lead to multiple solutions, for which the output of an iterative method may not converge. However, we can apply the proximal method described in [2] page 233. The idea behind the proximal method is to solve a series of problems each of which has a strictly concave objective function. The limit of the series approaches a single solution of the original problem. A detailed description of the proximal method is in [2]. We now introduce auxiliary variables $\overrightarrow{y}=\{y_i^k|i\in\{1,\ldots,I\},k\in\mathcal{N}(i)$, and $\overrightarrow{h}=\{h_{ij}^{lm}|i\neq j,i,j\in\{1,\ldots,I\},l\in\mathcal{N}(i,j),m\in\mathcal{N}(j,i)\}$. The intermediate optimization problem becomes:

$$\max_{\{\overrightarrow{x},\overrightarrow{g}\}\geq 0}\sum_{i=1}^{I}U_i(\sum_{k\in\mathcal{N}(i)}x_i^k+\sum_{\substack{i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}g_{ij}^{lm}) - \sum_{\substack{i\\ k\in\mathcal{N}(i)}}\frac{c_i}{2}(x_i^k-y_i^k)^2-\sum_{\substack{i,i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}\frac{d_i}{2}(g_{ij}^{lm}-h_{ij}^{lm})^2 \tag{3}$$

subject to:

$$\{\overrightarrow{x},\overrightarrow{g}\}\in\mathcal{WS}.$$

c_i and d_i are positive constants for each i. In the following, we focus on the dual of the intermediate maximization problem.

Since the Slater condition holds (see for reference [4]), there is no duality gap between the primal and the dual problems. Hence, we can use the duality approach to solve the problem.

Associate Lagrange multiplier λ_n with each link n, and μ_{ij}^{lm} with each PICC represented by graph G_{ij}^{lm}. Also, let $\overrightarrow{\lambda}$ and $\overrightarrow{\mu}$ be two column vectors with elements λ_n and μ_{ij}^{lm}, respectively. The Lagrange function of the above primal intermediate problem is:

$$\begin{aligned}L(\overrightarrow{x},\overrightarrow{g},\overrightarrow{\lambda},\overrightarrow{\mu},\overrightarrow{y},\overrightarrow{h}) &= \sum_{i=1}^{I}U_i(\sum_{k\in\mathcal{N}(i)}x_i^k+\sum_{\substack{i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}g_{ij}^{lm}) \\ &- \sum_{\substack{i\\ k\in\mathcal{N}(i)}}\frac{c_i}{2}(x_i^k-y_i^k)^2-\sum_{\substack{i,i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}\frac{d_i}{2}(g_{ij}^{lm}-h_{ij}^{lm})^2 \\ &- \sum_n\{\lambda_n\left\{\sum_{i=1}^{I}\sum_{k\in\mathcal{N}(i)}\mathcal{E}_i^{nk}x_i^k+\sum_{\substack{i,i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}F_{ij}^{nlm}g_{ij}^{lm}\right\}\} \\ &+ \sum_n\lambda_nC_n-\sum_{\substack{i,i<j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}(\mu_{ij}^{lm}g_{ij}^{lm})+\sum_{\substack{i,i<j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}(\mu_{ij}^{lm}g_{ji}^{ml})\end{aligned}$$

Given that $\sum_{\substack{i,i<j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}(\mu_{ij}^{lm}g_{ji}^{ml})=\sum_{\substack{i,j<i\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}}(\mu_{ji}^{ml}g_{ij}^{lm})$, the Lagrange function is separable as defined in Chapter 3 of [2] and we can write it as:

$$L(\overrightarrow{x},\overrightarrow{g},\overrightarrow{\lambda},\overrightarrow{\mu},\overrightarrow{y},\overrightarrow{h})=\sum_i B_i(\overrightarrow{x},\overrightarrow{g},\overrightarrow{\lambda},\overrightarrow{\mu},\overrightarrow{y},\overrightarrow{h})+\sum_n\lambda_nC_n.$$

Here,

$$B_i(\overrightarrow{x}, \overrightarrow{g}, \overrightarrow{\lambda}, \overrightarrow{\mu}, \overrightarrow{y}, \overrightarrow{h}) = U_i(\sum_{k\in\mathcal{N}(i)} x_i^k + \sum_{\substack{i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}} g_{ij}^{lm})$$

$$- \sum_{k\in\mathcal{N}(i)} \frac{c_i}{2}(x_i^k - y_i^k)^2 - \sum_{\substack{i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}} \frac{d_i}{2}(g_{ij}^{lm} - h_{ij}^{lm})^2$$

$$- \sum_k (\sum_n \mathcal{E}_i^{nk}\lambda_n) x_i^k - \sum_{\substack{i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}} (\sum_n F_{ij}^{nlm}\lambda_n) g_{ij}^{lm}$$

$$- \sum_{\substack{i<j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}} \mu_{ij}^{lm} g_{ij}^{lm} + \sum_{\substack{i>j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}} \mu_{ji}^{ml} g_{ij}^{lm}.$$

The objective function of the dual problem is

$$D(\overrightarrow{\lambda}, \overrightarrow{\mu}, \overrightarrow{y}, \overrightarrow{h}) = \max_{\{\overrightarrow{x}, \overrightarrow{g}\}\geq 0} L(\overrightarrow{x}, \overrightarrow{g}, \overrightarrow{\lambda}, \overrightarrow{\mu}, \overrightarrow{y}, \overrightarrow{h})$$

and the dual problem is:

$$\min_{\overrightarrow{\lambda}>0, \overrightarrow{\mu}} D(\overrightarrow{\lambda}, \overrightarrow{\mu}, \overrightarrow{y}, \overrightarrow{h}).$$

The dual optimization problem can be solved using the gradient method.

Based on the above we have the following distributed dual rate control algorithm (Algorithm $\mathcal{A}$).

Algorithm $\mathcal{A}$:

- Initialization phase:
  Find all paths between all sources and sinks. This can be done using any routing protocol that finds multiple paths as [19], [32]. After this, sources send control messages to every link n to set the values of H_{ij}^{nl} and $\mathcal{E}_i^{nk}$. Each s_i chooses the values of $y_i^k(0)$ and $h_{ij}^{lm}(0)$ randomly and each link sets the elements of $\overrightarrow{\mu}(0)$ and $\overrightarrow{\lambda}(0)$ to zero.
- Iteration phase: At the τ-th iteration:

  1) Fix $\overrightarrow{\lambda}(\tau, 0) = \overrightarrow{\lambda}(\tau)$, and $\overrightarrow{\mu}(\tau, 0) = \overrightarrow{\mu}(\tau)$, and perform the following steps sequentially for $r = 1, \ldots, K-1$.
     - Let $\{\overrightarrow{x}(\tau, r), \overrightarrow{g}(\tau, r)\} = \arg\max_{\{\overrightarrow{x}, \overrightarrow{g}\}\geq 0} L(\overrightarrow{x}, \overrightarrow{g}, \overrightarrow{\lambda}(\tau, r), \overrightarrow{\mu}(\tau, r), \overrightarrow{y}(\tau), \overrightarrow{h}(\tau))$.
       This can be computed in a distributed way at each source, because the L function is separable.
     - Update the dual variables at each link n by:

$$\lambda_n(\tau, r+1) = [\lambda_n(\tau, r) + \alpha_n(\sum_{\substack{i\\ k\in\mathcal{N}(i)}} \mathcal{E}_i^{nk} x_i^k(\tau, r) + \sum_{\substack{i,i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}} F_{ij}^{nlm} g_{ij}^{lm}(\tau, r) - C_n)]^+. \quad (4)$$

       Here, $[.]^+$ is a projection on $[0, \infty)$ and α_n is a positive step size. Also, $(\sum_{\substack{i\\ k\in\mathcal{N}(i)}} \mathcal{E}_i^{nk} x_i^k(\tau, r) + \sum_{\substack{i,i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}} F_{ij}^{nlm} g_{ij}^{lm}(\tau, r) - C_n)$ is the average queue length at link n, because we assume that queues empty out after every update.
     - Set

$$\mu_{ij}^{lm}(\tau, r+1) = \mu_{ij}^{lm}(\tau, r) + \beta_{ij}^{lm}(g_{ij}^{lm}(\tau, r) - g_{ji}^{ml}(\tau, r)) \qquad \forall i < j. \quad (5)$$

       This can be implemented at the destination, where β_{ij}^{lm} is a positive step size.

     It is worth noting that computing $x(\tau, r)$ and $g(\tau, r)$ needs the values of (i) $\sum_n F_{ij}^{nlm}\lambda_n(\tau, r) \quad \forall i < j, l, m$, which can be computed along the paths, (ii) $\mu_{ij}^{lm}(\tau, r)\ \forall i < j, l, m$, and (iii)$\mu_{ji}^{lm}(\tau, r)\ \forall i > j, l, m$. All of this information can be sent back to the source using an acknowledgment message.
  2) Let $\overrightarrow{\lambda}(\tau+1) = \overrightarrow{\lambda}(\tau, K)$ and $\overrightarrow{\mu}(\tau+1) = \overrightarrow{\mu}(\tau, K)$. Set:

$$y_i^k(\tau+1) = x_i^k(\tau, K) \qquad \forall i, k$$

     and

$$h_{ij}^{lm}(\tau+1) = g_{ij}^{lm}(\tau, K) \qquad \forall i < j, l, m.$$

Our assumption here is that the utility function is concave which does not hold for transmission scenarios like video streaming. In these cases Algorithm $\mathcal{A}$ can be modified as in [18] to achieve the optimal solution when the number of sessions is sufficiently large.

For sufficiently large K and sufficiently large number of iterations, $y(\tau)$ and $h(\tau)$ converges to the optimizing $\overrightarrow{x}$ and $\overrightarrow{g}$ for the original problem with the objective function in (2) and the constraints $\{\overrightarrow{x}, \overrightarrow{g}\} \in \mathcal{WS}$.

Theorem 2: As $K \longrightarrow \infty$, with the step sizes $(\alpha_n, \beta_{ij}^{lm})$ satisfying the following:
$Ł \max_n \alpha_n + 2\max_{\{i,j,l,m\}} \beta_{ij}^{lm} < 2\min_i(\min(c_i, d_i))$, where

$$Ł = \sum_{\substack{i\\ k\in\mathcal{N}(i)}} \mathcal{E}_i^{nk} + \sum_{\substack{i,i\neq j\\ l\in\mathcal{N}(i,j)\\ m\in\mathcal{N}(j,i)}} (F_{ij}^{nlm})^2,$$

Algorithm $\mathcal{A}$ converges to the optimal solution of (2) subject to $\{\overrightarrow{x}, \overrightarrow{g}\} \in \mathcal{WS}$.

The proof is provided in our online technical report [16]. For the case when K is bounded away from infinity, the convergence of Algorithm $\mathcal{A}$ is verified by simulations. Similar proofs to those in [23] can be used to rigorously prove the convergence of Algorithm $\mathcal{A}$ with fixed K and with noisy and delayed measurements. This makes Algorithm $\mathcal{A}$ suitable for practical implementation.

V. The Coding Scheme

The optimization problem and the solution described thus far allocate rates at each link so that the utility function can be optimized subject to $\{\overrightarrow{x}, \overrightarrow{g}\} \in \mathcal{WS}$. The next question is: what is the network coding scheme that can achieve the rate assignment? In this section, we propose the use of a scheme we call the Pairwise Random coding (PRC) scheme. From the construction of G_{ij}^{lm} in Section III, G_{ij}^{lm} is a graph satisfying the necessary and sufficient condition for pairwise network coding specified in Theorem 1. Since our discussion is based on fixed values of i, j, l, and m, for simplicity, we will use here $G_p = G_{ij}^{lm}$ as shorthand and assume that the PICC G_p is for the session pair (s_1, t_1) and (s_2, t_2). By normalizing over the rate g_{ij}^{lm} associated with G_{ij}^{lm}, we can assume that G_p has unit rate link capacity. We further assume that the messages for the unicast sessions (s_1,t_1), (s_2,t_2) are X_1, X_2, respectively.

A random network coding scheme is proposed in [11] for a single multicast session sending $\mathcal{R}$ messages $X_1, \ldots, X_{\mathcal{R}}$ to d receivers $t_1, \ldots, t_d$. These messages can be generated at one or many sources. For the case in which more than one source generates the messages, these sources can generate independent or linearly correlated messages in terms of $X_1, \ldots, X_{\mathcal{R}}$. The only assumption in [11] is that the min-cut max-flow between the source[1] and every receiver is no less than $\mathcal{R}$. The probability of success is defined as the probability that all sinks recover $X_1, \ldots, X_{\mathcal{R}}$ correctly. Some or all of the network code coefficients in [11] are chosen uniformly at random from a finite field of size q, where $q > d$, and the remaining code coefficients, if any, are fixed. The results in [11] show that if there exists a solution to the network connection problem with the same values for the fixed code coefficients, the success probability P_s is lower bounded by: $P_s \geq (1 - \frac{d}{q})^{|E|}$. We will show that with a simple modification, random network coding is applicable for pairwise inter-session network coding as well. Directly using random network coding for *pairwise inter-session network coding* without modification is infeasible, because the min-cut max-flow between the source and every sink may be strictly less than $\mathcal{R}$. (The value of $\mathcal{R}$ in this paper is two, since in each inter-session coding graph G_{ij}^{lm} only the two source-sink pairs (s_i, t_i) and (s_j, t_j) are considered.) Take Figure 1(b) for example, if we connect both s_1 and s_2 to a common source s^*, the min-cut max-flow between (s^*, t_1) is 1. Therefore, if random network coding is used for the inter-session network coding in Figure 1(b) t_1 will not be able to decode both X_1 and X_2. More specifically, we observe that for inter-session network coding problem when and only when either the paths in the set $\mathcal{P}_1 = \{P_{s_1,t_1}, P_{s_2,t_1}, Q_{s_1,t_1}\}$ or the paths in the set $\mathcal{P}_2 = \{P_{s_2,t_2}, Q_{s_1,t_2}, Q_{s_2,t_2}\}$ share the same link in G_p will the min-cut max-flow bound between the common source and the sinks be violated. Motivated by this observation, the PRC scheme is described in Figure 2 as follows.

for *each node v in the network* **do**
  for *each outgoing link n of v* **do**
    if ***A.*** $\{P_{s_1,t_1}, P_{s_2,t_1}, Q_{s_1,t_1}\}$ *share link n* **then**
      Decode X_1 and send it through n
    end
    if ***B.*** $\{P_{s_2,t_2}, Q_{s_1,t_2}, Q_{s_2,t_2}\}$ *share link n* **then**
      Decode X_2 and send it through n
    end
    if ***A.*** *is not satisfied and* ***B.*** *is not satisfied* **then**
      Randomly mix input messages to generate the output messages in n
    end
  end
end

Fig. 2: The PRC scheme

Theorem 3: Given that pairwise network coding is feasible on G_p, the probability that the PRC scheme is able to transmit X_1 and X_2 successfully through sessions (s_1, t_1) and (s_2, t_2), is lower bounded by $P_r(success) \geq (1 - \frac{4}{q})^{6|E|}$ $\forall q > 4$. The proof of this theorem is quit technical and long. It is provided in our online technical report [16].

[1]If there are more than one source, we can connect them to a virtual source by virtual links of appropriate capacity and the min-cut max flow now is between the virtual source and any other node in the network.

The PRC scheme can be implemented in a distributed way. Each node v needs only to know whether the paths in $\mathcal{P}_1$ (or $\mathcal{P}_2$) share the same outgoing link adjacent to v, which can be obtained during the initialization phase of Algorithm $\mathcal{A}$. Based on that piece of information, node v decides whether to perform decoding or random mixing. Furthermore, from *Theorem 3*, we can see that the success probability of PRC scheme approaches one when the size of the finite field is sufficiently large. Large field size can be achieved by moderate packet size. This is because we can have $q = 2^f$, where f is the packet size. In [6], it has been shown that even for moderately-sized $q = 2^{16}$ or $q = 2^8$, random linear coding approximates the optimal, centralized network coding schemes very well.

VI. Implementation Issues

In this section, we discuss several practical issues that may have impact on the implementation of our algorithm.

A. Collecting implicit costs

Each source s_i needs to collect $\sum_n \lambda_n F_{ij}^{nlm}$ for every PICC represented by G_{ij}^{lm}. Computing the sum of implicit costs $\sum_n \lambda_n F_{ij}^{nlm}$ can be done by transmitting a special control message ($\mathcal{SIC}$) through PICC G_{ij}^{lm} in each iteration according to the following rules:

- Each source sets $\mathcal{SIC}$ to zero and send it along all of its outgoing links.
- At link n, the received value of $\mathcal{SIC}$ in G_{ij}^{lm} plus $\lambda_n F_{ij}^{nlm}$ is the value of $\mathcal{SIC}$ that link n sends to the outgoing node in G_{ij}^{lm}.
- Every intermediate node forwards the sum of values in $\mathcal{SIC}$ (received from every incoming link in G_{ij}^{lm}) to only one outgoing link in G_{ij}^{lm}, and sends a value of zeros to all other outgoing links in G_{ij}^{lm}.

Following these rules, the sum of the values of $\mathcal{SIC}$ received at t_i and t_j in G_{ij}^{lm} is $\sum_n \lambda_n F_{ij}^{nlm}$. Both the $\mathcal{SIC}$ values received

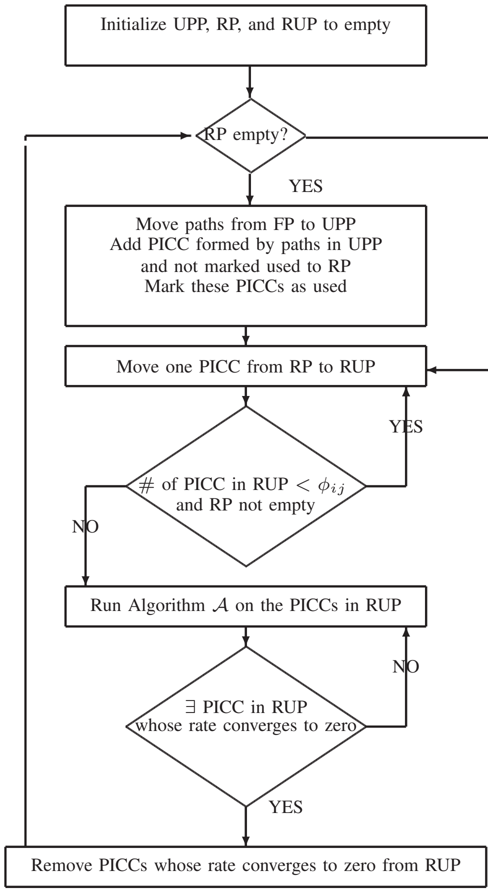


Fig. 3. Flow chart for the adaptive algorithm.

at t_i and t_j can be sent back to s_i with the acknowledgment message, and then s_i can obtain $\sum_n \lambda_n F_{ij}^{nlm}$.

B. Adaptive Algorithm

If all possible paths between $s_i, \forall i \in \{1, \ldots, I\}$ and $t_j, \forall j \in \{1, \ldots, I\}$ are included in the optimization simultaneously, the number of PICCs will be large. This means that Algorithm $\mathcal{A}$ will have high complexity, because every link n will have to set up a large number of F_{ij}^{nlm} and the control message communication overhead will be large. To reduce the complexity of the algorithm, we propose an adaptive scheme that works initially on a small number of paths. While Algorithm $\mathcal{A}$ is run on these paths, new paths will be added to the optimization problem one-by-one. This adaptive version reduces the communication and computation complexity of the original algorithm, because the rates on the insignificant PICCs converge to zero very quickly (generally after only a few iterations as observed in our simulations). Note that insignificant PICCs are those for which network coding provides no positive gain over routing. These PICCs can then be removed from the optimization problem, which reduces the variable space of the optimization problem. Figure 3 contains a detailed description of the above scheme with an adaptive path search mechanism.

In the flow chart in Figure 3, every source maintains Found Paths Pool (FP) and Used Paths Pool (UPP). Every source pair maintains Ready Pool (RP) and Running Pool (RUP). Every time new paths are found, the source puts them in FP which is executed in parallel to the steps in Figure 3. UPP contains all of the paths that the algorithm can use to form PICCs. When paths are moved to UPP, all possible PICCs that can be formed by the paths in the UPP and have not been examined yet, are put in RP. Each pair of sessions i and j is assigned a value ϕ_{ij}. ϕ_{ij} represents the maximum number of PICCs (for sessions i and j) that can be included in the maximization problem simultaneously. Every time insignificant PICCs (for sessions i and j) are removed from the optimization problem, s_i moves some of the PICCs (for sessions i and j) from RP to RUP. This is done in a way such that the total number of PICCs for i and j in RUP does not exceed ϕ_{ij}.

VII. Simulation Results

The objectives of the simulations are to verify the convergence of Algorithm $\mathcal{A}$, and to show the benefits of inter-session network coding in terms of throughput, fairness, and complexity.

A. Convergence

To study the convergence of Algorithm $\mathcal{A}$, we run simulations on the so called grail topology in Figure 1(b) with the utility function of each source s_i being $\log(R_i)$. As is obvious from the grail topology, there are three paths connecting (s_2-t_2), two paths connecting (s_1-t_2), two paths connecting (s_2-t_1), and one path connecting (s_1-t_1). Therefore, there are 36 possible PICCs. We assign the initial rates of each PICC randomly, and vary λ, μ and the number of proximal iterations K to test the speed of convergence of Algorithm $\mathcal{A}$. The results are plotted in Figures 4, 5, and 6.

The optimal solution for the grail topology is to assign unit rate to the PICC that uses $\mathcal{P} = \{P_{s_1,t_1} = s_1v_1v_2v_3v_4t_1$, $P_{s_2,t_2} = s_2v_1v_2t_2$, $P_{s_2,t_1} = s_2v_3v_4t_1\}$ and $\mathcal{Q} = \{Q_{s_1,t_1} = s_1v_1v_2v_3v_4t_1$, $Q_{s_2,t_2} = s_2v_3v_4t_2$, $Q_{s_1,t_2} = s_1v_1v_2t_2\}$, and zero rates to all of the other PICCs. These are the paths that satisfy Condition 2 in Theorem 1 as explained in section II. In Figures 4 and 5 we show the rates for the optimal PICC at both sources. Our algorithm converges even with a very small number of proximal iterations. As expected, increasing the step size up to a specific value will make the algorithm converge faster. In Figure 6, the rate of the insignificant PICC converges quickly to zero. A bigger topology in Figure 7 with 36 nodes and unit capacity links is used in our simulations. This topology has four unicast sessions. The convergence results for one of the optimal PICCs and one of the insignificant PICCs in the topology in Figure 7 are shown in Figure 8.

B. Gain and complexity

We compare Algorithm $\mathcal{A}$ with existing algorithms and quantify the benefits of inter-session network coding over routing solutions. The simulation is conducted on a graph depicted in Figure 7. For this topology, the TRLKM work [27] and its distributed implementation in [9] and [10] cannot

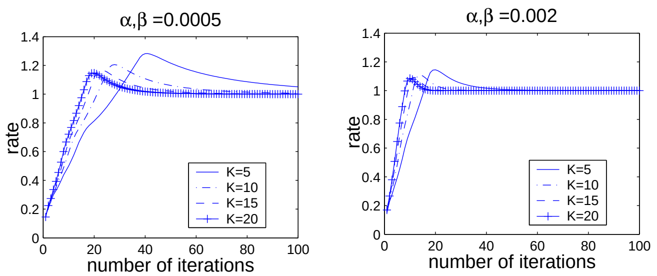


Fig. 4. Convergence results for s_1 in the grail topology with different step sizes and K, the number of proximal iterations. Here the rate corresponds to the optimal PICC.

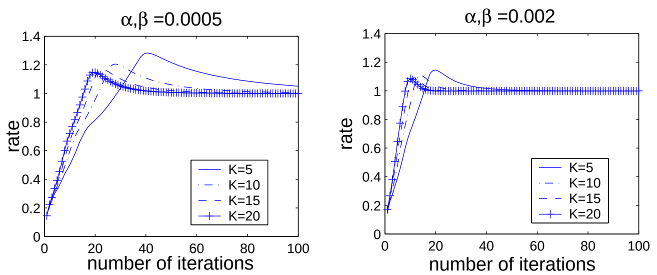


Fig. 5. Convergence results for s_2 in the grail topology with different step sizes and K, the number of proximal iterations. Here the rate corresponds to the optimal PICC.

realize any benefits of network coding and the performance of these algorithms is the same as that of routing since there is no butterfly structure available in Figure 7. It is worth noting that the distributed implementations of the TRLKM region in [9], [10] focus on stabilizing the given traffic load and do not focus on maximizing the utility function. We define the utility gain of inter-session network coding, $\mathcal{UG}$ as

$$\mathcal{UG} = \frac{Utility(Algorithm\mathcal{A}) - Utility(routing)}{Utility(routing)}.$$

We denote the total throughput of the network when the optimal utility is achieved under the $\mathcal{WS}$ and the routing regions by $\sum_i R_i(\mathcal{WS})$ and $\sum_i R_i(Routing)$, respectively. The throughput gain, $\mathcal{TG}$ is defined as

$$\mathcal{TG} = \frac{\sum_i R_i(\mathcal{WS}) - \sum_i R_i(Routing)}{\sum_i R_i(Routing)}.$$

We evaluate the gains of Algorithm $\mathcal{A}$ using different utility functions presented in [3] and [25]. The first type of utility function is $\log_2(\gamma + R_i)$, where γ is a constant in the range $[0, 1]$. The second type of utility function is of the form $\frac{R_i^{1-\sigma}}{1-\sigma}$, where σ is a constant in the range $(0, 1)$. The results are shown in Figures 9 and 10. Algorithm $\mathcal{A}$ outperforms both routing and TRLKM on this topology. Moreover, the largest throughput gain happens when fairness is the design criteria for the network, i.e, when γ is small and when σ is large. The proposed rate control algorithm indeed enhances fairness.

In terms of computational complexity between Algorithm $\mathcal{A}$ and the existing TRLKM method [27], the path based Algorithm $\mathcal{A}$ solves a maximization problem of 2,328 variables

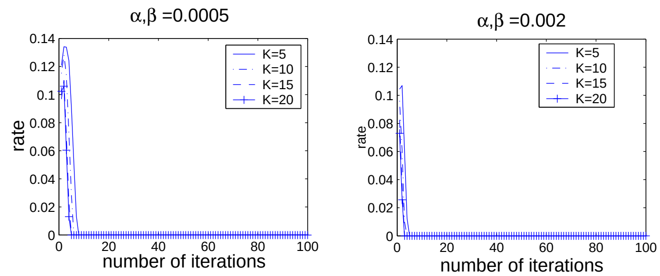


Fig. 6. Convergence results in the grail topology with different step sizes and K, the number of proximal iterations. Here the rate corresponds to the insignificant PICC.

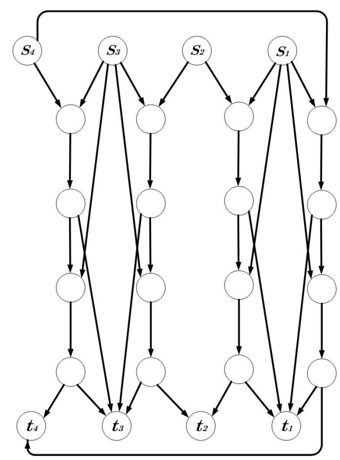


Fig. 7. Topology contains four source-sink pairs

and 36 constraints in a distributed way for the topology in Figure 7, while the pattern-search-based TRLKM optimization problem [27] has more than 31,104 variables and 31,104 constraints. In sum, the flexible choice of utility functions, decentralized rate control capability, superior performance in terms of utility/throughput gains, and manageable complexity with an adaptive path search mechanism, demonstrates the efficacy of the path-based Algorithm $\mathcal{A}$.

VIII. Conclusion

In this paper we develop a distributed rate control algorithm for the multiple-unicast-sessions problem. The algorithm supports rates in the $\mathcal{WS}$ capacity region that allows for inter-session network coding. We also propose a distributed pairwise random coding (PRC) scheme suitable for online implementation. Our algorithm improves both throughput and fairness among flows in information networks. Our results provide insights to network protocol designers on how to incorporate inter-session network coding in the currently implemented routing and congestion control protocols.

References

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

[2] D. Bertsekas and J. N. Tsitsikalis, *Parallel and distributed computation: Numerical methods*, Athena Scientific, 1997.

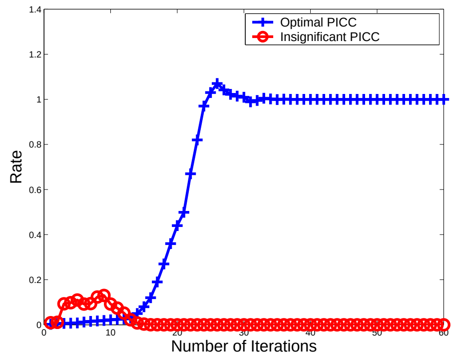


Fig. 8. Convergence results for the topology in Figure 7 with $\alpha = 0.01$, $\beta = 0.01$, and the number of proximal iterations $K = 5$. One of the optimal PICCs and one of the insignificant PICCs are examined.

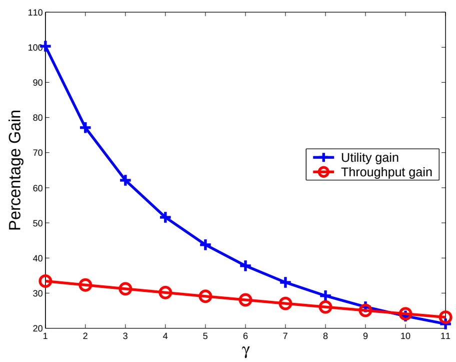


Fig. 9. Gain for objective function $\sum_i \log_2(\gamma + R_i)$ with different values of γ

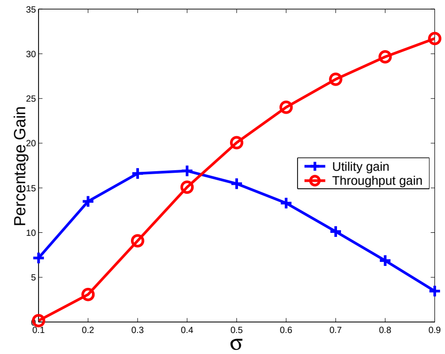


Fig. 10. Gain for objective function $\sum_i \frac{R_i^{1-\sigma}}{1-\sigma}$ with different values of σ

[3] T. Bonald and L. Massoulie, *Impact of fairness on internet performance*, in proceedings of ACM Sigmetrics, Cambridge, MA, June 2001.

[4] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[5] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, *Optimization based rate control for multicast with network coding*, in proceedings of IEEE INFOCOM, Anocharage, AK, May 2007.

[6] P. Chou, Y. Wu, , and K. Jain, *Practical network coding*, in proceedings of 41st Allerton Conf. Communication, Control and Computing, oct 2003.

[7] S. Deb and R. Srikant, *Congestion control for fair resource allocation in networks with multicast flows*, IEEE Trans. on Networking (2004), 274–285.

[8] R. Dougherty, C. Freiling, and K. Zeger, *Insufficiency of linear coding in network information flow*, IEEE Trans. on Information Theory **51** (2005), 2745–2759.

[9] A. Eryilmaz and D. S. Lun, *Control for inter-session network coding*, in proceedings of Netcod, San Diego, Jan 2007.

[10] T. Ho, Y. Chang, and K. J. Han, *On constructive network coding for multiple unicasts*, 44th Allerton Conference on Communication, Control and Computing, monticello, IL, Sept 2006.

[11] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, and D. Karger, *A random linear network coding approach to multicast*, IEEE/ACM Trans. on Information Theory Theory **52 (10)** (2006), 4413–4430.

[12] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, *Polynomial time algorithms for multicast network code construction*, IEEE Trans. on Information Theory **51** (2005), 1973–1982.

[13] K. Kar, S. Sarkar, and L. Tassiulas, *Optimization based rate control for multipath sessions*, Technical Report No. 2001-1, Institute for Systems Research, University of Maryland, 2001.

[14] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, *XORs in the air: Practical wireless network coding*, in proceedings of ACM SIGCOM, Pisa, Italy, Sept 2006.

[15] F. P. Kelly, A. Maulloo, and D. Tan, *Rate control in communication networks: Shadow prices, proportional fairness and stability*, Journal of the Operational Research Society **49** (1998), 237–252.

[16] A. Khreishah, C.-C. Wang, and N. B. Shroff, *Optimization based rate control for communication networks with inter-session network coding*, http://web.ics.purdue.edu/~akhreish/RCTR07.pdf, 2007.

[17] R. Koetter and M. Medard, *Beyond routing: An algebraic approach to network coding*, in proceedings of INFOCOM, New York, June 2002.

[18] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, *Non-convex optimization and rate control for multi-class services in the internet*, IEEE/ACM Trans. on Networking **13 (4)** (2005), 827–840.

[19] R. Leung, J. Liu, E. Poon, C. Chan, and B. Li, *MP-DSR: A QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks*, in proceedings of IEEE LCN, Tampa, FL, Nov 2001.

[20] R. Li, R. W. Yeung, and N. Cai, *Linear network coding*, IEEE Trans. on Information Theory **49(2)** (2003), 371–381.

[21] Z. Li and B. Li, *Network coding in undirected networks*, in proceedings of CISS, Princeton, NJ, March 2004.

[22] X. Lin and N. B. Shroff, *An optimization based approach for quality-of-service routing in high-bandwidth networks*, IEEE/ACM Trans. on Networking (2006).

[23] ———, *Utility maximization for communication networks with multi-path routing*, IEEE Trans on Automatic Control (2006).

[24] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, *Achieving minimum-cost multicast: A decentralized approach based on network coding*, in proceedings of IEEE INFOCOM, Miami, FL, March 2005.

[25] J. Mo and J. Walrand, *Fair end-to-end window-based congestion control*, IEEE/ACM Trans. on Networking 8-5 (2000).

[26] S. Sengupta, S. Rayanchu, and S. Banerjee, *An analysis of wireless network coding for unicast sessions: The case for coding-aware routing*, in Proceedings of IEEE INFOCOM, Anocharage, AK, May 2007.

[27] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Medard, *Network coding for multiple unicasts: An approach based on linear optimization*, IEEE International Symposium on Information Theory .Seatle, 2006.

[28] C.-C. Wang and N. B. Shroff, *Beyond the butterfly – a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions*, in proceedings IEEE International Symposium on Information Theory, Nice, France, June 2007.

[29] Y. Wu, M. Chiang, and S.-Y. Kung, *Distributed utility maximization for network coding based multicasting: a critical cut approach*, in Proceedings 2nd workshop on Network Coding, Theory, and Applications, Boston, Apr 2006.

[30] Y. Wu, K. Jain, and S.-Y. Kung, *A unification of network coding and tree-packing (routing) theorems*, IEEE Trans. on Information Theory **52** (2006), 2398–2409.

[31] Y. Wu and S.-Y. Kung, *Distributed utility maximization for network coding based multicasting: a shortest path approach*, IEEE J. on Selected Areas in Communications, special issue on Nonlinear Optimization of Communication Systems **24** (2006), 1475–1488.

[32] W. T. Zaumen and J. J. Garcia-Luna-Aceves, *Loop-free multipath routing using generalized diffusing computations*, in proceedings of IEEE INFOCOM, San Francisco, CA, March 1998.