Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Distributed Minimum Cost Multicasting with Lossless Source Coding and Network Coding

Tao Cui, Tracey Ho, and Lijun Chen

Abstract— **In this paper, we consider minimum cost lossless source coding for multiple multicast sessions. Each session comprises a set of correlated sources whose information is demanded by a set of sink nodes. We propose a distributed end-to-end algorithm which operates over given multicast trees, and a back-pressure algorithm which optimizes routing and coding over the whole network. Unlike other existing algorithms, the source rates need not be centrally coordinated; the sinks control transmission rates across the sources. With random network coding, the proposed approach yields completely distributed and optimal algorithms for intra-session network coding. We prove the convergence of our proposed algorithms. Some practical considerations are also discussed. Experimental results are provided to complement our theoretical analysis.**

I. Introduction

In wireless networks such as sensor networks, information sources may be correlated. Independent data compression and transmission, without considering such correlations, is not an optimal strategy in terms of efficient use of the wireless spectrum. Higher efficiency can be obtained by using distributed lossless source coding techniques [1]. Network coding, a generalization of routing to allow nodes to perform algebraic operations on packets, is another technique that can significantly improve network performance [2]. In this paper, we consider, for a distributed network scenario, joint optimization of source coding, rate allocation, multicast network coding, and scheduling.

Some aspects of this problem are considered in a number of related works. In [3], joint optimization of Slepian-Wolf coding and routing is considered. For the multicast case, the approach involves finding a minimum cost Steiner tree. The scenario of multi-sink is considered in [4], where a sub-optimal distributed scheme is proposed which also requires information exchange between sources. An algorithm to find the minimum cost subgraph for joint source coding and network coding is proposed in [5], for the case of two sources. In [6], a practical lossless source coding scheme is proposed for a network with a single sink. Even though Slepian-Wolf coding is distributed, the optimization problems in [3]–[5] still require coordination among the sources to guarantee that the source rates lie in the Slepian-Wolf region. Moreover, the designs in [3] need central controllers to solve the routing problem. Therefore, the algorithms in [3]–[5] cannot be fully distributed.

In [7], dynamic algorithms for intra-session network coding and scheduling, for both correlated and uncorrelated sources, are proposed based on *back-pressure*. The algorithms are concerned only with rate stabilization, without consideration of costs. Interestingly, our proposed session scheduling policy for cost optimization is similar to the *reverse back-pressure policy* for correlated sources in [7] though they are obtained differently. The policy in [7] is obtained by intuition, whereas our policy is obtained from the utility maximization framework, providing a principled derivation.

Motivated by the utility optimization framework developed for TCP congestion control, see, e.g., [8], we consider the problem of minimizing an aggregate cost measure defined in terms of the flow rates at the links, with the source rates being constrained within the Slepian-Wolf region. Solved in a centralized way, our problem is a convex optimization problem with a polynomial time solution. However, since centralized solutions are less desirable in practice, we develop and analyze a distributed algorithm. Unlike other approaches from the literature [3]–[6], our approach admits a fully distributed implementation for both the case with given multicast trees and the case without given multicast trees. As in [8], our algorithms can be interpreted as distributed primal-dual algorithms over the network to minimize the total cost. Both algorithms use primal-dual subgradient algorithms. More importantly, our algorithms remove the source coordination requirement in the previous work [3]–[5]. With random network coding [9], all our algorithms can be implemented in a fully distributed manner. Our proposed algorithms can be readily extended to multicasting without network coding.

II. Preliminaries

A. Network Model

Consider a network, denoted by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, with a set $\mathcal{N}$ of nodes and a set $\mathcal{L}$ of directed links. We denote a link either by a single index l or by the directed pair (i, j) of nodes it connects. A set of multicast sessions $\mathcal{M}$ is transmitted through the network. Each session $m \in \mathcal{M}$ is associated with a set $\mathcal{S}_m \subset \mathcal{N}$ of sources and a set of $\mathcal{T}_m \subset \mathcal{N}$ of sinks. In session m, each source $s \in \mathcal{S}_m$ multicasts x^{ms} bits to all the sinks in $\mathcal{T}_m$. By flow conservation,

$$\sum_{j:(i,j)\in\mathcal{L}} g_{i,j}^{mst} - \sum_{j:(j,i)\in\mathcal{L}} g_{j,i}^{mst} = \sigma_i^{ms}, \forall i \in \mathcal{N}, \tag{1}$$

where $\sigma_i^{ms} = x^{ms}$ if $i = s$, $\sigma_i^{ms} = -x^{ms}$ if $i = t$, $\sigma_i^{ms} = 0$ otherwise, and $g_{i,j}^{mst}$ is the flow rate on link (i, j) from source $s \in \mathcal{S}_m$ to sink $t \in \mathcal{T}_m$ in session m.

This work has been supported in part by DARPA grant N66001-06-C-2020, Caltech's Lee Center for Advanced Networking, a gift from Microsoft Research, and NSF through grant CNS-0435520.

T. Cui, T. Ho, and L. Chen are with Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA, USA 91125. Email: {taocui, tho}@caltech.edu, chen@cds.caltech.edu

1-4244-1498-9/07/$25.00 ©2007 IEEE.

Authorized licensed use limited to: CALIFORNIA INSTITUTE OF TECHNOLOGY. Downloaded on April 20,2010 at 21:01:32 UTC from IEEE Xplore. Restrictions apply.

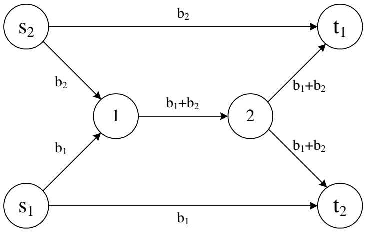

Fig. 1. The butterfly network.

We consider a wireless network model which subsumes wireline networks as a special case. We assume a static topology where each link $l \in \mathcal{L}$ has a fixed capacity c_l bits per second when it is active. Each link $l \in \mathcal{L}$ is associated with a cost function $\beta_l(\cdot)$, which is a strictly convex, monotonically increasing function of the total flow on each link (the case of linear cost function is solved in [10]). Such cost functions arise naturally when the cost is, e.g., latency or congestion. We adopt the primary interference model as in [11] without considering broadcast advantage. Under this model, any feasible schedule corresponds to a matching of $\mathcal{G}$. Let $\mathcal{E}$ denote the set of all matchings, indexed by e. We represent a matching e as an $|\mathcal{L}|$-dimensional rate vector, $\mathbf{r}^e$ with $r_l^e = c_l$ if $l \in e$, and $r_l^e = 0$ otherwise. The feasible rate region $\mathbf{\Pi}$ at the link layer is then defined as the convex hull of all the matching rate vectors or equivalently

$$\mathbf{\Pi} \triangleq \left\{ \mathbf{r} : \mathbf{r} = \sum_{e \in \mathcal{E}} a_e \mathbf{r}^e,\, a_e \geq 0, \sum_{e \in \mathcal{E}} a_e = 1 \right\}. \qquad (2)$$

B. Network Coding

Each node is allowed to perform algebraic operations on received packets in network coding. It has been shown that the ability of the network to transfer information can be significantly improved [2]. The butterfly example in Fig. 1 shows benefit of using network coding. In this paper, we simply assume that coding is done only across packets of the same session. With this setting, we define $f_{i,j}^m$ as the physical flow of session m on link (i,j). By the flow sharing property of network coding and capacity constraints, we have the following two constraints

$$g_{i,j}^{mst} \leq f_{i,j}^m, \quad \forall (i,j) \in \mathcal{L},\, m \in \mathcal{M},\, s \in \mathcal{S}_m,\, t \in \mathcal{T}_m, \quad (3)$$
$$\sum_{m \in \mathcal{M}} f_{i,j}^m \leq r_{i,j}, \qquad \forall (i,j) \in \mathcal{L}, \qquad (4)$$

where $r_{i,j}$ belongs to a rate vector within $\mathbf{\Pi}$ in (2).

To ensure fully distributed cross-layer design, we use distributed random network coding [9], where for each node the data on outgoing links are random linear combination of the data on incoming links.

C. Lossless Source Coding

We consider multiterminal lossless source coding [1], where data cannot be distorted in the compression process. Data of correlated sources are compressed and jointly decoded at sinks. An important scenario for multiterminal source coding is distributed source coding, where correlated sources compress data separately without communicating each other. For the case of n sources, let $\mathcal{X} = \{X_1, \ldots, X_n\}$ be the set of sources. Slepian-Wolf coding is a technique for distributed lossless source coding. The achievable rate region for distributed lossless source coding is defined by [1]

$$\sum_{X_i \in \mathcal{S}} R_i \geq H(\mathcal{S}|\mathcal{X}\backslash\mathcal{S}),\, \forall \mathcal{S} \subseteq \mathcal{X}, \qquad (5)$$

where R_i denotes the data rate of X_i. It is shown in [9] that in general networks, (5) still holds by using random network coding. Recent advance in practical source code designs has shown that (5) is achievable [12] by using error correcting codes such as low-density parity-check codes.

III. Distributed End-End Control Algorithm with Given Multicast Trees

In this section, a set of multicast trees is given for each source in every session. We only consider a single multicast tree for each source as it can be obtained by using protocols such as distance vector multicast routing protocol [13]. To simplify notation, we consider the case where overlapping links of different trees of a session have disjoint sets of downstream destinations, thus allowing coding to occur on all overlapping links.

Let T^{ms} denote the multicast tree for source s in session m. Each tree T^{ms} contains a set $\mathcal{L}_{ms} \subseteq \mathcal{L}$ of links, which defines a $|\mathcal{L}| \times 1$ vector ξ^{ms} whose l-th entry is given by

$$\xi_l^{ms} = \begin{cases} 1, & \text{if } l \in T^{ms}, \\ 0, & \text{otherwise.} \end{cases} \qquad (6)$$

We apply intra-session network coding. Similar to (3) and (4), we have the following two constraints

$$\xi_l^{ms} x^{ms} \leq f_l^m, \qquad \forall l \in \mathcal{L},\, m \in \mathcal{M},\, s \in \mathcal{S}_m, \qquad (7)$$
$$\sum_m f_l^m \leq r_l, \qquad \forall l \in \mathcal{L}. \qquad (8)$$

where $r_l \in \mathbf{\Pi}$, and (7) is due to the assumption that overlapping links of different trees of a session have disjoint sets of downstream destinations. For the case without network coding, (7) and (8) become $\sum_m \xi_l^{ms} x^{ms} \leq r_l$ and the solution is modified accordingly.

A. Problem Formulation

The basic problem is to minimize the total network cost subject to the source rates being in the Slepian-Wolf region (5) and the rate constraints (7)-(8). Thus, we need to solve the following optimization problem[12]

$$\begin{aligned} \min_{x^{ms}, f_l^m, r_l} \quad & \sum_{l \in \mathcal{L}} \beta_l \left(\sum_{m \in \mathcal{M}} f_l^m \right) \\ \text{subject to} \quad & \xi_l^{ms} x^{ms} \leq f_l^m,\, \forall l \in \mathcal{L},\, m \in \mathcal{M},\, s \in \mathcal{S}_m \\ & \sum_{m \in \mathcal{M}} f_l^m \leq r_l,\, \forall l \in \mathcal{L},\, \mathbf{r} \in \mathbf{\Pi} \\ & \sum_{s \in \mathcal{S}} x^{ms} \geq H(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S}),\, \forall \mathcal{S} \subseteq \mathcal{S}_m,\, m \in \mathcal{M}. \end{aligned} \qquad (9)$$

Note that we also have a constraint $x^{ms} \geq 0$ but we assume it implicitly in the following for simplicity. (9) can be readily

[1] Note that in (9) we assume the separation of source coding and channel coding, which is suboptimal in general. Thus our algorithms in this paper are optimal only for separate source coding and channel coding.

[2] In practice, due to the use of finite block length codes and the oscillation of the subgradient algorithm, $H(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S})$ in (9) should be replaced by $H(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S}) + \epsilon$, where $\epsilon > 0$. In the following, for brevity, we solve (9).

Authorized licensed use limited to: CALIFORNIA INSTITUTE OF TECHNOLOGY. Downloaded on April 20,2010 at 21:01:32 UTC from IEEE Xplore. Restrictions apply.

transformed into a convex optimization problem and solved in polynomial time with a central controller. However, a distributed algorithm is preferred in practice. Note that we always assume that (9) is feasible in this section. How to deal with infeasible source rates can be found in [10].

B. Distributed Algorithm

Note that the last set of constraints in (9) makes all the source rates coupled. Even though distributed source coding can be applied independently at each source, solving (9) directly still needs the coordination of sources such that the source rates lie in the Slepian-Wolf region. For this reason, the algorithms in [3], [5] cannot be fully distributed, and the algorithm in [4] is suboptimal and requires explicit information exchange between sources. We solve these problems by a distributed receiver-driven source coding algorithm. To recover the source rates, we employ a primal-dual subgradient algorithm.

Instead of solving (9), we consider an equivalent problem by replacing the last constraint in (9) with the following constraints

$$\begin{aligned}&\sum_{s\in\mathcal{S}} y^{mst} \geq H(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S}),\ \forall\mathcal{S}\subseteq\mathcal{S}_m,\ m\in\mathcal{M},\ t\in\mathcal{T}_m,\\&y^{mst}\leq x^{ms},\ \forall l\in\mathcal{L},\ m\in\mathcal{M},\ s\in\mathcal{S}_m,\ t\in\mathcal{T}_m,\end{aligned}\tag{10}$$

where y^{mst} is the new variable introduced at each sink $t\in\mathcal{T}_m$ corresponding to x^{ms}.

The Lagrangian dual function obtained by relaxing the first, the second constraints in (9) and the last constraints in (10) can be decomposed into four subproblems[3]

$$\phi_1(q)=\min_y\sum_{m,s,t} q_t^{ms}y^{mst},\ \text{s.t.}\sum_{s\in\mathcal{S}} y^{mst}\geq H(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S}),\tag{11}$$

$$\phi_2(p,q)=\min_x\sum_{m,s} x^{ms}\left(\sum_l p_l^{ms}\xi_l^{ms}-\sum_t q_t^{ms}\right),\tag{12}$$

$$\phi_3(p,\lambda)=\min_f\sum_l \beta_l\left(\sum_m f_l^m\right)-\sum_{l,m}\left(\sum_s p_l^{ms}-\lambda_l\right)f_l^m,\tag{13}$$

$$\phi_4(\lambda)=\max_r\sum_l \lambda_l r_l,\ \text{subject to}\ \mathbf{r}\in\mathbf{\Pi}.\tag{14}$$

where p_l^m is the Lagrange multiplier introduced at link l for session m, λ_l is the Lagrange multiplier introduced at link l, and q_t^{ms} is the Lagrange multiplier introduced at sink t for source s in session m. The first subproblem is minimum cost virtual lossless source coding [3] at each sink. The second subproblem is rate allocation. The third one is the joint network coding and session scheduling. The fourth one is the link scheduling, which does not exist in wireline networks. Thus, by dual decomposition, the flow optimization problem decomposes into separate "local" optimization problems of application, transport, and network/data link layers, respectively. The four subproblems interact through Lagrange multipliers p, q, λ. We first solve the dual subproblems (11)-(14) by assuming fixed dual variables p, λ, and q.

Source coding: We can further decompose (11) into $|\mathcal{T}_m|$ virtual minimum cost lossless source coding problems[4] for each sink in session m. Let $N_m=|\mathcal{S}_m|$ denote the number of sources in session m. For sink $t\in\mathcal{T}_m$, we need to solve

$$\min_y\sum_s q_t^{ms}y^{mst},\ \text{subject to}\sum_{s\in\mathcal{S}} y^{mst}\geq H(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S}).\tag{15}$$

Due to the duality between Slepian-Wolf and multiple access channels, following the approach in [14], it can be easily shown that the region defined by the constraint in (15) is a contra-polymatroid [14]. From Lemma 3.3 in [14], greedy algorithm solves (15) optimally. Let π^* be any permutation of $\mathcal{S}_m$ such that $q_t^{m\pi^*(1)}\leq q_t^{m\pi^*(2)}\leq\cdots\leq q_t^{m\pi^*(N_m)}$. The solution of (15) is given by

$$\begin{aligned}y^{m\pi^*(1)t}&=H\left(\pi^*(1)\right),\\y^{m\pi^*(2)t}&=H\left(\pi^*(2)|\pi^*(1)\right),\end{aligned}\tag{16}$$

$$\begin{aligned}&\vdots\\y^{m\pi^*(N_m)t}&=H\left(\pi^*(N_m)|\pi^*(N_m-1),\ldots,\pi^*(1)\right).\end{aligned}\tag{17}$$

Note that [3] also gives a similar solution to (15).

Rate allocation: If we solve (12) directly as in [15], the solution to (12) is either zero or unbounded as the objective function in (12) is not strictly convex in x^{ms}. The source rate cannot be recovered. We thus apply the primal subgradient algorithm to resolve this problem. Let $x^{ms}(\tau)$ and $x^{ms}(\tau+1)$ denote the source rates at time τ and $\tau+1$ respectively. In the primal subgradient algorithm, we update x^{ms} by using a primal subgradient algorithm as

$$x^{ms}(\tau+1)=\left[x^{ms}(\tau)-\epsilon_\tau\left(\sum_l p_l^{ms}\xi_l^{ms}-\sum_t q_t^{ms}\right)\right]^+,\tag{18}$$

where ϵ_τ is a positive scalar stepsize, and $[\cdot]^+$ denotes the projection onto $\mathbb{R}^+$. Source rate is adjusted not only according to the aggregate dual variables $\sum_l p_l^{ms}\xi_l^{ms}$ over the multicast tree T^{ms} but also according to the aggregate dual variables $\sum_t q_t^{ms}$ due to virtual source coding, which are fed back from the sinks in session m.

The source rates are guaranteed to lie inside the Slepian-Wolf region through dual variable q_t^{ms} without the coordination of sources. Each source compresses data according to rate $x^{ms}(\tau+1)$ in (18) by using Slepian-Wolf coding or randomized linear network coding. This rate allocation mechanism is an end-to-end control mechanism.

Session scheduling and network coding: For each link l, let m_l^* be the multicast session, which has the maximum aggregate link dual variables, i.e., $m_l^*=\arg\max_m\sum_s p_l^{ms}$, and define $w_l=\sum_s p_l^{m_l^*s}$. The optimal solution of (13) can be obtained as

$$f_l^m=\begin{cases}f_l^*, & \text{if } m=m_l^*,\\ 0, & \text{otherwise},\end{cases}\tag{19}$$

where f_l^* is the maximizer of

$$\max_f\left(w_l-\lambda_l\right)f-\beta_l\left(f\right).\tag{20}$$

To see why (19) is correct, if we transmit packets at rate f' for a non-minimum congestion price session m', we can shift the rate f' from m' to m_l^* to make the objective function of

[3] In (9), we actually replace $\max_s\{\xi_l^{ms}x^{ms}\}\leq f_l^m$ with $|\mathcal{S}_m|$ constraints $\xi_l^{ms}x^{ms}\leq f_l^m$, $\forall s\in\mathcal{S}_m$, and introduce $|\mathcal{S}_m|$ multipliers p_l^{ms} at each link. When $|\mathcal{S}_m|$ is large, we can instead approximate $\max_i(x_i)$ with $(\sum_i x_i^n)^{1/n}$ for large n.

[4] We call it virtual source coding problem as we do not perform source coding at sinks.

Authorized licensed use limited to: CALIFORNIA INSTITUTE OF TECHNOLOGY. Downloaded on April 20,2010 at 21:01:32 UTC from IEEE Xplore. Restrictions apply.

(20) smaller. As $\beta_l(\cdot)$ is strictly convex and monotonically increasing, the optimal solution of (20) is obtained as

$$f_l^* = \begin{cases} \beta_l'^{-1}(w_l - \lambda_l) & \text{if } w_l > \lambda_l, \\ 0 & \text{if } w_l \le \lambda_l. \end{cases} \tag{21}$$

For each link l, a random linear combination of packets from all the sources in session m_l^* is sent at the rate of f_l^*.

Link scheduling: From (2), $\mathbf{\Pi}$ is a polyhedron. The objective function $\sum_l \lambda_l r_l$ in (14) is also linear. It is known that the maximum of (14) is attained at an extreme point of $\mathbf{\Pi}$, which corresponds to a matching on graph $\mathcal{G}$. Therefore, (14) reduces to finding a maximum weighted matching in $\mathcal{G}$. Distributed approximation algorithm for maximum weighted matching in [16] can be applied to solve (14) distributedly, which achieves a factor of 2 approximation. In the following, we assume that (14) is solved optimally.

Dual variable update: After solving the dual subproblems (11)-(13), we update the dual variables using subgradient method. Let $p(\tau)$, $\lambda(\tau)$, and $q(\tau)$ denote the dual variables p, λ, and q at time τ. By the subgradient method, each link l updates its dual variable p with respect to source s in session m according to

$$p_l^{ms}(\tau+1) = \left[p_l^{ms}(\tau) + \gamma_\tau\left(\xi_l^{ms} x^{ms}(\tau) - f_l^m(\tau)\right)\right]^+, \tag{22}$$

and updates its dual variable λ as

$$\lambda_l(\tau+1) = \left[\lambda_l(\tau) + \gamma_\tau\left(\sum_m f_l^m(\tau) - r_l(\tau)\right)\right]^+, \tag{23}$$

and each sink t updates its dual variable with respect to source s in session m according to

$$q_t^{ms}(\tau+1) = \left[q_t^{ms}(\tau) + \gamma_\tau\left(y^{mst}(\tau) - x^{ms}(\tau)\right)\right]^+, \tag{24}$$

where γ_τ is a positive scalar stepsize. Note that (22)-(24) are distributed and can be implemented by individual links using only local information.

By assuming perfect scheduling or solving (14) optimally, we have the following theorem.

Theorem: The primal-dual subgradient algorithm (16)-(24) converges to the optimal solution of (10) with diminishing stepsize.

Please refer to technical report [10] for the proof. For a constant stepsize, the primal-dual subgradient method converges within any given small neighborhood around the optimum, by choosing sufficiently small constant stepsize.

IV. Distributed Back-Pressure Algorithm without Given Multicast Trees

Our distributed algorithm in this section is based on back-pressure policy. Back-pressure rate control algorithm is a kind of hop-by-hop rate control algorithm.

A. Distributed Algorithm

As in Section III, we minimize the total cost of the network. But in this section, we consider the rate constraints (1)-(4) and the lossless source coding constraint (5). Even though distributed lossless source coding can be applied independently at each source, solving this problem directly still needs the coordination of sources due to the constraint (5). Similar to (9), we first transform this problem into an equivalent problem, which can be solved distributedly. We consider the following problem

$$\begin{aligned} \min_{x,y,g,f,r} \quad & \sum_{(i,j)\in\mathcal{L}} \beta_{i,j}\left(\sum_{m\in\mathcal{M}} f_{i,j}^m\right) \\ \text{subject to} \quad & \sum_{j:(i,j)\in\mathcal{L}} g_{i,j}^{mst} - \sum_{j:(j,i)\in\mathcal{L}} g_{j,i}^{mst} = \eta_i^{mst}, \forall i \in \mathcal{N}, \\ & \sum_{s\in\mathcal{S}_m} g_{i,j}^{mst} \le f_{i,j}^m, \forall (i,j)\in\mathcal{L}, m\in\mathcal{M}, t\in\mathcal{T}_m, \\ & \sum_{m\in\mathcal{M}} f_{i,j}^m \le r_{i,j}, \forall (i,j)\in\mathcal{L}, \mathbf{r}\in\mathbf{\Pi} \\ & \sum_{s\in\mathcal{S}} y^{mst} \ge H(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S}), \forall \mathcal{S}\subseteq\mathcal{S}_m, m\in\mathcal{M}, t\in\mathcal{T}_m, \end{aligned} \tag{25}$$

where

$$\eta_i^{mst} = \begin{cases} x^{ms}, & \text{if } i = s \\ -y^{mst}, & \text{if } i = t \\ 0, & \text{otherwise,} \end{cases} \tag{26}$$

and y^{mst} is the new variable introduced at each sink $t \in \mathcal{T}_m$ corresponding to x^{ms}. Due to the flow conservation constraint, any feasible solution of (25) satisfies $y^{mst} = x^{ms}$. Therefore, (25) solves the problem optimally.

The dual problem to (25) by relaxing only the first and the third constraints in (25) can be decomposed into four subproblems

$$\phi_1(p) = \min_y -\sum_{m,s,t} p_t^{mst} y^{mst}, \text{ s.t. } \sum_{s\in\mathcal{S}} y^{mst} \ge H(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S}), \tag{27}$$

$$\phi_2(p) = \min_x \sum_{m,s} x^{ms}\left(\sum_t p_s^{mst}\right), \tag{28}$$

$$\begin{aligned} \phi_3(p,\lambda) = \max_{g,f,r} & \sum_{i,m,s,t} p_i^{mst}\left(\sum_{j:(i,j)\in\mathcal{L}} g_{i,j}^{mst} - \sum_{j:(j,i)\in\mathcal{L}} g_{j,i}^{mst}\right) \\ & - \sum_{i,j} \beta_{i,j}\left(\sum_{m\in\mathcal{M}} f_{i,j}^m\right) - \sum_{i,j} \lambda_{i,j} \sum_m f_{i,j}^m, \\ & \text{subject to } \sum_s g_{i,j}^{mst} \le f_{i,j}^m, \end{aligned} \tag{29}$$

$$\phi_4(\lambda) = \max_r \sum_{i,j} \lambda_{i,j} r_{i,j}, \text{ subject to } \mathbf{r}\in\mathbf{\Pi}, \tag{30}$$

where p_i^{mst} is the Lagrange multiplier introduced at node i for source s and sink t in session m, and $\lambda_{i,j}$ is the Lagrange multiplier introduced at link (i,j). Note that if $p_t^{mst} > 0$ in (27), $\phi_1(p)$ is unbounded. Thus, we must have $p_t^{mst} \le 0$. The four subproblems interact through dual variables p, λ. We first solve the dual subproblems (27)-(30) by assuming fixed dual variables p, λ.

Source coding: The source coding problem (27) can be solved as in Section III. We omit here for brevity.

Rate allocation: As the objective function in (25) is not strictly convex in x^{ms}, similar to Section III, we also use primal subgradient algorithm to recover the source. Let $x^{ms}(\tau)$ and $x^{ms}(\tau+1)$. We update x^{ms} as

$$x^{ms}(\tau+1) = \left[x^{ms}(\tau) - \epsilon_\tau\left(\sum_t p_s^{mst}\right)\right]^+, \tag{31}$$

where ϵ_τ is a positive scalar stepsize. In (31), source rate is adjusted according to the aggregate dual variables $\sum_t p_s^{mst}$

Authorized licensed use limited to: CALIFORNIA INSTITUTE OF TECHNOLOGY. Downloaded on April 20,2010 at 21:01:32 UTC from IEEE Xplore. Restrictions apply.

generated locally at the source node. Compared with (18), (31) does not need the dual variables fed back from all the links over the multicast tree T^{ms} and all the sinks in session m, which suggests a small delay. The source rates are guaranteed to lie inside the Slepian-Wolf region through $\sum_t p_s^{mst}$ without the coordination of sources. Penalty function method can be applied similar to Section III.

Each source compresses data according to rate x^{ms} in (31) by using randomized linear network coding.

Session scheduling and network coding: Note that (29) is equivalent to the following problem

$$\begin{aligned} &\max_{g,f} \sum_{i,j,m,s,t} g_{i,j}^{mst}\left(p_i^{mst} - p_j^{mst}\right) - \sum_{i,j} \beta_{i,j}\left(\sum_m f_{i,j}^m\right) \\ &\qquad\qquad - \sum_{i,j} \lambda_{i,j} \sum_m f_{i,j}^m, \\ &\text{subject to } \sum_s g_{i,j}^{mst} \le f_{i,j}^m, \qquad\qquad (32) \\ &= \max_{g,f,r} \sum_{i,j} \left(\sum_m f_{i,j}^m \sum_t \max_s \left[p_i^{mst} - p_j^{mst}\right]^+ \right. \\ &\qquad\qquad \left. -\beta_{i,j}\left(\sum_m f_{i,j}^m\right) - \lambda_{i,j} \sum_m f_{i,j}^m \right), \end{aligned}$$

where the last equality comes from the fact that $\sum_s g_{i,j}^{mst}\left(p_i^{mst} - p_j^{mst}\right)$, subject to $\sum_s g_{i,j}^{mst} \le f_{i,j}^m$ is a linear programming, we can always choose an extreme point solution, i.e.,

$$g_{i,j}^{mst} = \begin{cases} f_{i,j}^m, & \text{if } s = \left(s^{mt}\right)^*, \text{ and } p_i^{mst} - p_j^{mst} \ge 0, \\ 0, & \text{otherwise,} \end{cases} \qquad (33)$$

where $\left(s^{mt}\right)^* = \arg\max_s \left(p_i^{mst} - p_j^{mst}\right)$. For each link (i,j), let $m_{i,j}^*$ be the multicast session, which has the maximum aggregate differential link prices, i.e., $m_{i,j}^* = \arg\max\limits_m \sum\limits_t \max\limits_s \left[p_i^{mst} - p_j^{mst}\right]^+$, and define $w_{i,j} = \max_m \sum_t \max_s \left[p_i^{mst} - p_j^{mst}\right]^+$. As in (13), the solution to (32) is

$$f_{i,j}^m = \begin{cases} f_{i,j}^* & \text{if } m = m_{i,j}^*, \\ 0, & \text{otherwise,} \end{cases} \qquad (34)$$

where $f_{i,j}^*$ is the maximizer of

$$\max_f (w_{i,j} - \lambda_{i,j}) f - \beta_{i,j}(f). \qquad (35)$$

If $\beta_{i,j}(\cdot)$ is a strictly convex function, the optimal solution solving (35) is

$$f_{i,j}^* = \begin{cases} \beta_{i,j}^{\prime -1}\left(w_{i,j} - \lambda_{i,j}\right) & \text{if } w_{i,j} > \lambda_{i,j}, \\ 0 & \text{if } w_{i,j} \le \lambda_{i,j}. \end{cases} \qquad (36)$$

For each link (i,j), a random linear combination of packets from sources $\left(s^{m_{i,j}^* t}\right)^*$, $t \in \mathcal{T}_{m_{i,j}^*}$, in session $m_{i,j}^*$ is sent at the rate of $f_{i,j}^*$. This is equivalent to solving (29) by the following assignment

$$g_{i,j}^{mst} = \begin{cases} f_{i,j}^*, \text{ if } m = m_{i,j}^*,\ s = \left(s^{mt}\right)^*, \text{ and } p_i^{mst} - p_j^{mst} > 0, \\ 0, \qquad \text{otherwise.} \end{cases} \qquad (37)$$

Link scheduling: Solving (30) is similar to that in Section III.

Dual variable update: Let $p(\tau)$ and $\lambda(\tau)$ denote the dual variables p, λ at time τ, respectively. After solving (27)-(30), by the subgradient method, each node i updates its dual variable p with respect to source s and sink t in session m according to

$$\begin{aligned} &p_i^{mst}(\tau+1) = \\ &\begin{cases} p_i^{mst}(\tau) + \gamma_\tau\Big(x^{ms}(p(\tau)) \\ \qquad - \sum_j g_{i,j}^{mst}(p(\tau)) + \sum_j g_{j,i}^{mst}(p(\tau))\Big), & \text{if } i = s, \\ \Big[p_i^{mst}(\tau) + \gamma_\tau\Big(- y^{mst}(p(\tau)) \\ \qquad - \sum_j g_{i,j}^{mst}(p(\tau)) + \sum_j g_{j,i}^{mst}(p(\tau))\Big)\Big]_-, & \text{if } i = t, \\ p_i^{mst}(\tau) + \gamma_\tau\Big(\sum_j g_{j,i}^{mst}(p(\tau)) \\ \qquad - \sum_j g_{i,j}^{mst}(p(\tau))\Big), & \text{otherwise,} \end{cases} \end{aligned} \qquad (38)$$

and every link (i,j) updates its dual variable λ as

$$\lambda_{i,j}(\tau+1) = \left[\lambda_{i,j}(\tau) + \gamma_\tau\left(\sum_m f_{i,j}^m(p(\tau),\lambda(\tau)) - r_{i,j}(\lambda(\tau))\right)\right]^+, \qquad (39)$$

where γ_τ is positive scalar stepsize, and $[\cdot]_-$ denotes the projection onto $\mathbb{R}^-$. After node i updates its congestion price, it passes $p_i^{mst}(\tau+1)$ to all its neighbors for next time slot source coding, rate allocation, scheduling and network coding.

Note that our algorithm (31)-(39) only requires nodes to communicate with direct neighbors. Thus, our design is a hop-by-hop control mechanism. Note that the above session scheduling component uses back-pressure to do optimal scheduling, similarly to [7], [15]. However, the dual variable p_i^{mst} in our algorithm is negative, while in traditional back-pressure p_i^{mst} is positive and it can be interpreted as the queue length. We interpret the negative dual variable p_i^{mst} as the virtual queue length at each node. Physically, the negative p_i^{mst} or virtual queue length indicates how many bits are still required such that the sinks can decode the compressed data in the end. Interestingly, the virtual source coding component is also similar to that in [7], and a similar virtual queue concept is also proposed in [7]. Our algorithm is obtained from the utility maximization framework while that in [7] is obtained intuitively. Another difference of our algorithm from those in [7], [15] is that in [7], [15] node i immediately sends packets to node j whenever the accumulated queue length difference $\sum\limits_t \max\limits_s \left[p_i^{mst} - p_j^{mst}\right]^+$ is greater than zero, while our algorithm requires that $\sum\limits_t \max\limits_s \left[p_i^{mst} - p_j^{mst}\right]^+$ is greater than $\lambda_{i,j}$. Clearly, when $\lambda_{i,j} = 0$ or no link cost, our policy reduces to that of the original back-pressure policy. By using network coding, we circumvent the difficulty of finding a minimum cost Steiner tree as in [3]. Moreover, our algorithm is fully distributed without the coordination of sources.

The same convergence result holds as that in Section III. Other practical issues are discussed in [10].

V. Experimental Results

We consider the butterfly network shown in Fig. 1. For simplicity, we assume that there is only one session and the network is a wireline network. Two sources s_1 and s_2 multicast correlated information to t_1 and t_2. All links

Authorized licensed use limited to: CALIFORNIA INSTITUTE OF TECHNOLOGY. Downloaded on April 20,2010 at 21:01:32 UTC from IEEE Xplore. Restrictions apply.

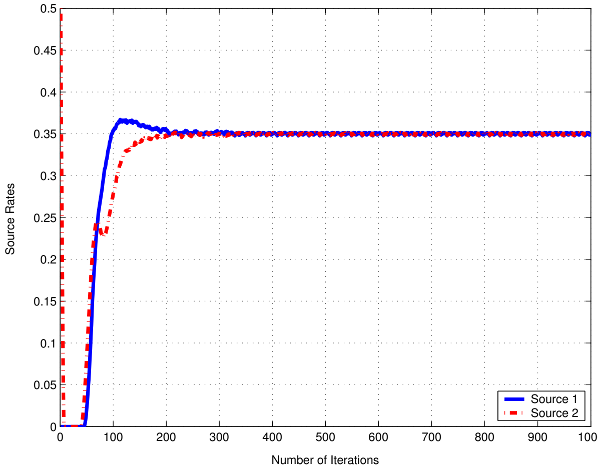

Fig. 2. The evolution of source rates versus the number of iterations of the distributed algorithm with given multicast trees and stepsizes $\epsilon = 0.05$ and $\gamma = 0.05$ Fig. 1.

have unit capacity, and have the same linear cost function $\beta(x) = 0.5x$. We assume that $H(s_1|s_2) = H(s_2|s_1) = 0.2$ and $H(s_1) = H(s_2) = 0.5$. We do not consider the delay due to dual variable feedback. All the dual variables are initialized randomly. It is easy to show that the optimal source rates for the two sources are 0.35 and 0.35. We use primal subgradient algorithm to update the source rate.

We first show the results for the case with given multicast trees. The multicast tree for source s_1 is chosen as $\{(s_1, 1), (1, 2), (2, t_2), (s_1, t_1)\}$, and for source s_2 is chosen as $\{(s_2, 1), (1, 2), (2, t_1), (s_2, t_2)\}$. Fig. 2 shows the evolution of source rates versus the number of iterations of the distributed algorithm with given multicast trees and fixed stepsizes $\epsilon = 0.05$ and $\gamma = 0.05$. We see that both source rates converge quickly to a neighborhood of the optimal rates and oscillate around them since we have chosen a constant stepsize. This also illustrates the validity of Theorem 1. This oscillating behavior mathematically results from the non-differentiability of the dual function.

We next consider distributed algorithm without given multicast trees. Fig. 3 shows the evolution of source rates versus the number of iterations in this case with fixed stepsizes $\epsilon = 0.05$ and $\gamma = 0.05$. Comparing Fig. 3 with Fig. 2, we find that back-pressure algorithm converges more slowly than end-to-end algorithm. The example in this section is simple. In general, the cost of back-pressure algorithm may be less than that using algorithm with given multicast trees since the capacity region for the latter case is a subset of the capacity region for the former case.

VI. Conclusion

We presented fully distributed algorithms for lossless source coding and rate allocation for multiple multicast sessions with correlated sources. Based on utility maximization framework, distributed algorithms for the cases with and without multicast trees were proposed. Intra-session network coding avoids the NP-hardness of finding a minimum cost Steiner tree. The sinks control transmission rates across the sources via local updates that propagate back to the sources.

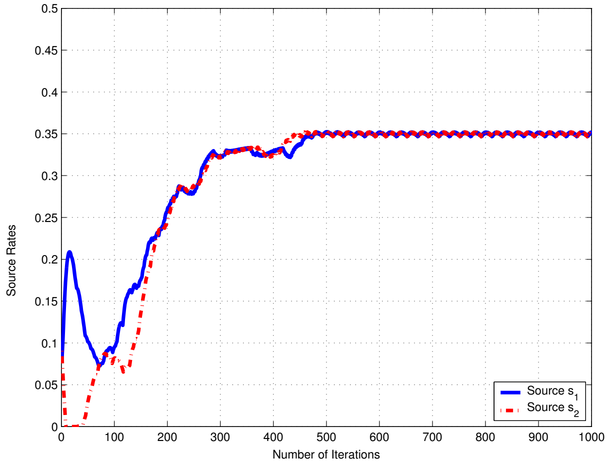

Fig. 3. The evolution of source rates versus the number of iterations of the distributed algorithm without given multicast trees and stepsizes $\epsilon = 0.05$ and $\gamma = 0.05$ in Fig. 1.

This is in contrast to existing algorithms where source rate control is achieved via centralized coordination among sources. It is of practical interest to study the implementation of our algorithms. Finally, designing universal distributed source codes that can work well for any correlation statistics is of interest in its own right.

References

[1] T. Cover and J. Thomas, *Elements of Information Theory*, 1991.

[2] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[3] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, “Networked Slepian-Wolf: Theory, algorithms and scaling laws,” *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4057–4073, Dec. 2005.

[4] K. Yuen, B. Li, and B. Liang, “Distributed data gathering in multi-sink sensor networks with correlated sources,” in *Proc. of IFIP Networking*, May 2006, pp. 868–879.

[5] A. Lee, M. Médard, K. Haigh, S. Gowan, and P. Rubel, “Minimum-cost subgraphs for joint distributed source and network coding,” in *Proc. of Workshop on Network Coding, Theory and Applications*, 2007.

[6] J. Liu, M. Adler, D. Towsley, and C. Zhang, “On optimal communication cost for gathering correlated data through wireless sensor networks,” in *Proc. of ACM Mobicom*, 2006, pp. 310–321.

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

[8] F. Kelly, A. Maulloo, and D. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, March 1998.

[9] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 4413–4430, Oct. 2006.

[10] T. Cui, T. Ho, and L. Chen, “Distributed minimum cost multicasting with lossless source coding and network coding,” Caltech, Tech. Rep., March 2007.

[11] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, “Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks,” in *Proc. of IEEE Infocom*, Apr. 2006.

[12] A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Commun. Lett.*, vol. 6, pp. 440–442, Oct. 2002.

[13] D.Waitzman, S. Deering, and C. Partridge, “Distance vector multicast routing protocol,” RFC 1075, Nov.1988.

[14] D. Tse and S. Hanly, “Multiaccess fading channels-part I: Polymatroid structure, optimal resource allocation and throughput capacities,” *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 2796–2815, Nov. 1998.

[15] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, “Rate control for multicast with network coding,” in *Proc. of IEEE Infocom*, 2007.

[16] J.-H. Hoepman, “Simple distributed weighted matchings,” eprint cs.DC/0410047, Oct. 2004.

Authorized licensed use limited to: CALIFORNIA INSTITUTE OF TECHNOLOGY. Downloaded on April 20,2010 at 21:01:32 UTC from IEEE Xplore. Restrictions apply.