Distributed Optimization for Wireless Networks with Inter-Session Network Coding

Ziyu Shao and Shuo-Yen Robert Li
Institute of Network Coding, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
Email: zyshao@inc.cuhk.edu.hk, bobli@ie.cuhk.edu.hk

***Abstract*—Network coding has been applied widely in wireless networks. In this paper, we focus on the cross-layer optimization of wireless networks with multiple unicast sessions and network coding. By exploiting broadcast advantage and one hop opportunistic listening, we develop a fully distributed solution including primal-dual flow control, Markov chain based hyperlink scheduling, session-decomposition coding scheme and session scheduling. We further study the convergence property of the distributed solution without time-scale separation assumption. We show the convergence to optimum with some time-dependent step sizes and update intervals. We also show the convergence to the bounded neighborhood of optimum with constant step sizes and constant update intervals. Our numerical evaluations validate the analytical results. We emphasis that though the analysis for both cases is quite involved, the resulting distributed solutions are actually simple to implement.**

I. Introduction

Network coding, introduced in [1, 2], allows intermediate nodes to perform coding operation instead of routing (store and forward). It is well known that intra-session network coding [3, 4, 5] is suboptimal for multi-session networks and inter-session network coding is necessary [6]. However, performing optimal inter-session network coding is complex and largely open [6]. Therefore, many suboptimal schemes for inter-session network coding [7, 8, 9, 10, 11, 12, 13, 14] are proposed. These schemes can be further classified into two subcategories: search the local butterfly structure and exploit wireless one hop coding opportunity. For the butterfly based approach, the achievable rate region was studied in [14] and the associated back-pressure algorithms were studied in [9, 10]. The one hop coding opportunity approach was proposed in COPE protocol [11]. Then it was followed by numerous work including characterization of throughput region with centralized scheduling [13], energy efficient scheduling with opportunistic coding [8], the power and throughput trade-off between multicast and unicast [7], and binary XOR based pairwise inter-session network coding [12].

Despite these exciting results on network coding, there are still many challenges when we try to solve the cross-layer optimization problem of wireless networks with multiple unicast sessions and network coding. In fact, this problem is NP-hard and hard to approximate even in a centralized manner. The hardness comes from two aspects. On one hand, the number of feasible independent sets (possible combinations of non-interfering wireless links) can be exponentially large. On the other hand, the number of feasible codes can be exponentially large. For example, some linear code constructions for multi-session network coding [6]. In this paper, we do not discuss the optimal code construction problem. Instead, we adopt low complexity network codes. Our main results and contributions are listed as follows:

- We provide a distributed solution for cross-layer optimization of wireless networks with multiple unicast sessions and network coding. Our solution achieves network utilities arbitrarily close to optimum. In contrast, previous distributed algorithms [8, 12] either achieve a small faction of maximum utility or assume special interference models. Further, without the time-scale separation assumption, we show the convergence to optimum with diminishing step sizes and increasing update intervals. We also show the convergence to the bounded neighborhood of optimum with constant step sizes and constant update intervals.

Due to page limits, all proofs can be found in our technical report [15].

II. Cross-layer Optimization of Multi-session Wireless Networks with Network Coding

A. Problem Setting and Formulation

We adopt one suboptimal inter-session network coding strategy, which exploits one-hop coding opportunity where each coded packet is decoded at the immediate next-hop node and searches the local-butterfly structures by session decomposition approach [8]. This approach decomposes multiple unicast sessions into a superposition of multicast and unicast sessions.

We exploit the *broadcast advantage* [13] of wireless networks, i.e., a single packet transmission might be overheard by a subset of receiver nodes within range of the transmitter. Therefore, we model a multi-hop wireless network as a hypergraph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of hyperlinks. A hyperlink $(i, J) \in \mathcal{L}$ represents a one-hop broadcast transmission, where $i \in \mathcal{N}$ is the transmitter and $J \subseteq \mathcal{N}$ is the set of receivers. Any two hyperlinks either interfere with each other, or they can be activated simultaneously. We adopt the conflict graph model to represent this interference relationship of hyperlinks. The conflict graph $\mathcal{G}_c = (\mathcal{N}_c, \mathcal{L}_c)$ is constructed as follows: for each hyperlink $l \in \mathcal{L}$, there is a corresponding vertex $v_l \in \mathcal{N}_c$. Then if the hyperlinks l and $l' \in \mathcal{L}$ interfere with each other, there is an edge between v_l and $v_{l'}$ An independent set of the conflict graph $\mathcal{G}_c$ is a set of non-adjacent nodes in $\mathcal{G}_c$, representing a set of hyperlinks that can be activated simultaneously without interfering with each other.

Now we consider a wireless network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ with a set of unicast sessions. Each unicast session is associated with one commodity, including one source and one destination. The set of commodities is denoted by S, where within each commodity $s \in S$, a source s sends packets to its destination t_s (unicast) at a rate of x_s. Let $\boldsymbol{x} = [x_s, s \in S]^T$ denote the vector of unicast rates. Then after session decomposition, we have a set of multicast sessions $\mathcal{M}$ and a set of new unicast sessions $\mathcal{V}$, where $|\mathcal{V}| = |S|$. Any new unicast session $v_s \in \mathcal{V}$ includes only one commodity $s \in S$, while any multicast session $m \in \mathcal{M}$ includes multiple commodities belonging to set S. There are some heuristic methods to exploit one-hop coding opportunity and construct set $\mathcal{M}$ [11, 13]. Here we assume that $\mathcal{M}$ has been constructed and fixed.

For each hyperlink $(i, J) \in \mathcal{L}$, let $f^{v_s}_{iJj}$ denote the flow rate of unicast session v_s over (i, J) and is intended to node $j \in J$; let f^{ms}_{iJj} denotes the flow rate of multicast session m (including the commodity s) over (i, J) and is intended to node $j \in J$. Let $\mathcal{H}$ be the set of all independent sets over the corresponding conflict hypergraph $\mathcal{G}_c$. $\mathbf{p} = [p_h, h \in \mathcal{H}]^T$ is denoted to be the vector of probability (or time fraction) of all independent sets. $\lambda^h_{i,J}$ denotes the capacity of hyperlink (i, J) within the independent set h. Given the hyperlink $(i, J) \in \mathcal{L}$, let z^m_{iJ} denote the physical rate of multicast session m over (i, J), and $z^{v_s}_{iJ}$ denote the physical rate of unicast session v_s over (i, J), $\forall\, m \in \mathcal{M}, v_s \in V$. Let $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function and event $i = s$ denote that node i is the source of commodity s, then we consider the following master utility maximization problem

$$\mathbf{MP}: \max_{\boldsymbol{x},\boldsymbol{f},\boldsymbol{z},\boldsymbol{p}\geq 0} \quad \sum_{s\in S} U_s(x_s) \tag{1}$$

$$\text{s.t.}\;\; x_s \mathbb{1}_{i=s} + \sum_{j\in\mathcal{N}} \sum_{\{i|(j,I)\in\mathcal{L}, i\in I\}} (\sum_{m\in\mathcal{M}} f^{ms}_{jIi} + f^{v_s}_{jIi}) \leq \sum_{\{J|(i,J)\in\mathcal{L}\}} \sum_{j\in J} (\sum_{m\in\mathcal{M}} f^{ms}_{iJj} + f^{v_s}_{iJj}), \forall\, i \in \mathcal{N} - \{t_s\}, s \in S \tag{2}$$

$$\sum_{j\in J} f^{v_s}_{iJj} \leq z^{v_s}_{iJ}, \;\; \forall\,(i, J) \in \mathcal{L}, v_s \in \mathcal{V} \tag{3}$$

$$\sum_{j\in J} f^{ms}_{iJj} \leq z^m_{iJ} \;\; \forall\,(i, J) \in \mathcal{L}, s \in S, m \in \mathcal{M} \tag{4}$$

$$\sum_{m\in\mathcal{M}} z^m_{iJ} + \sum_{v_s\in\mathcal{V}} z^{v_s}_{iJ} \leq \sum_{h\in\mathcal{H}} p_h\, \lambda^h_{iJ}, \;\; \forall\,(i, J) \in \mathcal{L} \tag{5}$$

$$\sum_{h\in\mathcal{H}} p_h = 1 \tag{6}$$

where the constraint (2) comes from the property of multi-commodity flow balance, the constraints (3)-(6) come from the flow-sharing property of network coding and time-sharing capacity-constraints [8].

Solving the master problem $\mathbf{MP}$ (1) is very challenging because the scheduling subproblem is NP-hard in general. To see that, we relax the first set of inequality constraints (2) in problem $\mathbf{MP}$ with Lagrange multipliers $\{r^s_i\}, s \in S, i \in \mathcal{N} - \{t_s\}$. $r^s_{t_s} = 0$, $\forall s \in S$ and $\boldsymbol{r} = [r^s_i, s \in S, i \in \mathcal{N}]^T$ is the vector. Then by Lagrange dual decomposition method and focus on the equivalent dual problem of $\mathbf{MP}$, we can obtain the following scheduling subproblem:

$$\mathbf{SSP}: \max_{\boldsymbol{p}\geq 0} \sum_{h\in\mathcal{H}} p_h \sum_{(i,J)\in\mathcal{L}} \lambda^h_{iJ}\, w_{iJ} \tag{7}$$

$$\text{s.t.} \;\; \sum_{h\in\mathcal{H}} p_h = 1.$$

and $\forall (i, J) \in \mathcal{L}$,

$$w_{iJ} = \max(w^1_{iJ}, w^2_{iJ}) \tag{8}$$

$$w^1_{iJ} = \max_{v_s\in\mathcal{V}} \max_{j\in J} (r^s_i - r^s_j)_+ \tag{9}$$

$$w^2_{iJ} = \max_{m\in\mathcal{M}} \sum_{s:s\in m} \max_{j\in J} (r^s_i - r^s_j)_+ \tag{10}$$

where $[\cdot]_+ \triangleq \max(\cdot, 0)$. This is the Maximum Weight Independent Set (MWIS) problem, where the associated weight is $\sum_{(i,J)\in\mathcal{L}} \lambda^h_{iJ}\, w_{iJ}$ for each independent set $h \in \mathcal{H}$. Since the MWIS problem is NP-hard [16], solving scheduling subproblem is very challenging. We utilize the Markov approximation framework [16] to approximately solving this problem in a distributed way.

B. Markov Approximation

First, we approximate the max function by the log-sum-exp function, i.e.,

$$\max_{h\in\mathcal{H}} \sum_{(i,J)\in\mathcal{L}} \lambda^h_{iJ}\, w_{iJ} \approx \frac{1}{\beta} \log(\sum_{h\in\mathcal{H}} \exp(\beta \sum_{(i,J)\in\mathcal{L}} \lambda^h_{iJ}\, w_{iJ})). \tag{11}$$

where β is a positive constant. In this way, we are implicitly solving an approximated version of the problem $\mathbf{SSP}$ (7), off by an entropy term $-\frac{1}{\beta}\sum_{h\in\mathcal{H}} p_h \log p_h$, shown as follows:

$$\mathbf{SSP}-\beta: \max_{\boldsymbol{p}\geq 0} \sum_{h\in\mathcal{H}} p_h \sum_{(i,J)\in\mathcal{L}} \lambda^h_{iJ}\, w_{iJ} - \frac{1}{\beta} \sum_{h\in\mathcal{H}} p_h \log p_h$$

$$\text{s.t.} \;\; \sum_{h\in\mathcal{H}} p_h = 1.$$

and the corresponding (unique) optimal solution is

$$p_h = \frac{\exp(\beta \sum_{(i,J)\in\mathcal{L}} \lambda^h_{iJ}\, w_{iJ})}{\sum_{h'\in\mathcal{H}} \exp(\sum_{(i,J)\in\mathcal{L}} r^{h'}_{iJ}\, w_{iJ})}, \forall h \in \mathcal{H} \tag{12}$$

Now we are solving a problem close to the original problem $\mathbf{MP}$ (1) as $\beta \to \infty$:

$$\mathbf{MP}-\beta: \max_{\boldsymbol{x},\boldsymbol{f},\boldsymbol{g},\boldsymbol{p}\geq 0} \quad \sum_{s\in S} U_s(x_s) - \frac{1}{\beta} \sum_{h\in\mathcal{H}} p_h \log p_h \tag{13}$$

$$\text{s.t.} \;\; (2)-(6)$$

By standard Lagrange dual decomposition method, we show that solving the problem $\mathbf{MP}-\beta$ (13) is equivalent to finding the saddle point of the following problem

$$\mathbf{DP}-\beta : \min_{\boldsymbol{r}\geq 0} \max_{\boldsymbol{x}\geq 0} L_{\beta}(\boldsymbol{x},\boldsymbol{r}) = \sum_{s\in S}[U_s(x_s) - x_s r_s^s] + \frac{1}{\beta}\log[\sum_{h\in\mathcal{H}} \exp(\beta \sum_{(i,J)\in\mathcal{L}} \lambda_{iJ}^h w_{iJ})] \tag{14}$$

We explore algorithm design in the following subsections. The $p_h, h\in\mathcal{H}$ in (12) can be interpreted as the stationary distribution of a time reversible Markov chain, whose states are the independent sets in $\mathcal{H}$. We first discuss how to design and implement such a Markov chain in a distributed manner, then we design primal-dual algorithms to solve the problem $\mathbf{DP}-\beta$.

C. Design and Implementation of Markov Chain

We start by only allowing direct transitions between two "adjacent" states (independent sets) h and h' that differ by one and only one hyperlink. Note that doing so will not affect the stationary distribution for time-reversible Markov chains. By this design, the transition from h' to $h = h' \cup \{(i,J)\}$ corresponds to hyperlink (i,J) starting its transmission. Similarly, the transition from h to h' corresponds to hyperlink (i,J) finishing its on-going transmission.

Now we consider two states h and h', where $h = h' \cup \{(i,J)\}$. Recall that $\lambda_{iJ}^h = \lambda_{iJ}\mathbb{1}_{(i,J)\in h} = \lambda_{iJ}$. We set $q_{h,h'}$ to $\lambda_{iJ}^h = \lambda_{iJ}$. Then by detailed balance equation, we have

$$\begin{aligned} q_{h',h} &= \lambda_{iJ}^h \exp(\beta(\sum_{(i',J')\in\mathcal{L}} \lambda_{i'J'}^h w_{i'J'} - \sum_{(i',J')\in\mathcal{L}} \lambda_{i'J'}^{h'} w_{iJ})) \\ &= \lambda_{iJ}^h \exp\left(\beta\lambda_{iJ}^h w_{iJ}\right). = \lambda_{iJ}\exp\left(\beta\lambda_{iJ} w_{iJ}\right). \end{aligned}$$

To achieve transition rate $q_{h',h}$, the node i set a timer for hyperlink (i,J), which counts down according to an exponential distribution with rate $\lambda_{iJ}\exp\left(\beta\lambda_{iJ}w_{iJ}\right)$. When the timer expires, hyperlink (i,J) starts to transmit. During the count-down, if the node i senses that another interfering node is in transmission, node i will freeze its count-down process.

When the transmission is over, node i counts down according to the residual back-off time, which is still exponential distributed with the same rate, because of the memoryless property of exponential distributions.

The transition rate $q_{h,h'}$ can be achieved by node i setting its transmission time to follow exponential distribution with rate $\lambda_{i,J}$.

The corresponding pseudocode is shown in Algorithm 1.

We establish the following result:

Proposition 1. *Algorithm 1 in fact implements a time-reversible Markov chain with stationary distribution in (12).*

D. Distributed Solution

In this subsection, we provide a distributed solution including primal-dual flow control algorithm, scheduling and coding policy:

Algorithm 1 Implementation of Markov Chain

The following procedures run on each individual hyperlink independently. We focus on a particular hyperlink (i,J) with the capacity $\lambda_{i,J}$.

procedure INITIALIZATION((i,J))
    Obtains w_{iJ} based on back-pressure of hyperlink (i,J)
    index $\leftarrow 0$
    Invokes Procedure Wait-and-Transmit((i,J))
end procedure

procedure WAIT-AND-TRANSMIT((i,J))
    generates a timer T_{iJ} following an exponential distribution with rate $\lambda_{i,J}\exp\left(\beta\lambda_{i,J}w_{iJ}\right)$ respectively and begin counting down.
    while the timer T_{iJ} does not expire **do**
        if Senses the transmission of interfering hyperlinks **then**
            index $\leftarrow 1$
            break
        end if
    end while
    Terminates current countdown process
    if index = 1 **then**
        index $\leftarrow 0$
        Invokes Procedure Wait-and-Transmit((i,J))
    else
        Sets the transmit time to follow an exponential distribution with rate $\lambda_{i,J}$ and transmits
    end if
end procedure

Primal-Dual Flow Control: The primal-dual subgradient algorithm is given as follows:

$$\begin{cases} \dot{x}_s = \alpha_s[U_s^{'}(x_s) - r_s^s]_{x_s}^+, \forall s\in S \\ \dot{r}_i^s = k_i^s[sub_i^s]_{r_i^s}^+, \forall\, i\in\mathcal{N}-\{t_s\}, s\in S \\ \dot{r}_{t_s}^s = r_{t_s}^s = 0, \forall s\in S \end{cases}, \tag{15}$$

Where

$$\begin{aligned} sub_i^s &= x_s\mathbb{1}_{i=s} + \sum_{j\in\mathcal{N}} \sum_{\{i|(j,I)\in\mathcal{L}, i\in I\}} (\sum_{m\in\mathcal{M}} f_{jIi}^{ms} + f_{jIi}^{v_s}) \\ &- \sum_{\{J|(i,J)\in\mathcal{L}\}} \sum_{j\in J} (\sum_{m\in\mathcal{M}} f_{iJj}^{ms} + f_{iJj}^{v_s}), \forall i\in\mathcal{N}, s\in S \end{aligned}$$

, $k_i^s (i\in\mathcal{N}-\{t_s\}, s\in S)$ and $\alpha_s (s\in S)$ are positive constants.

Hyperlink Scheduling: According to Algorithm 1.

Session Scheduling and Coding: for each hyperlink $(i,J)\in\mathcal{L}$, let

$$v_{iJ} = \arg\max_{v_s\in V}\max_{j\in J}(r_i^s - r_j^s)_+ \tag{16}$$

$$m_{iJ} = \arg\max_{m\in\mathcal{M}}\sum_{s:s\in m}\max_{j\in J}(r_i^s - r_j^s)_+ \tag{17}$$

When $w_{iJ} = 0$, node i transmits NULL packets. When $w_{iJ} > 0$, if $w_{iJ} = w^1_{iJ}$, a unicast session v_{iJ} is chosen. Node i broadcasts a packet from commodity v_{iJ} to all receivers $j \in J$ at rate $\sum\limits_{h\in\mathcal{H}} p_h\ \lambda^h_{iJ}$.

Otherwise, $w_{iJ} = w^2_{iJ}$, a multicast session m_{iJ} is chosen. Node i codes a packet by XOR-ing together packets from all commodities in m_{iJ} (one packet per commodity), then broadcasts the coded packet to all receivers $j \in J$ at rate $\sum\limits_{h\in\mathcal{H}} p_h\ \lambda^h_{iJ}$.

Opportunistic Listening: for all hyperlinks $(i, J) \in \mathcal{L}$, let

$$j^s_{iJ} = \arg\max_{j\in J}(r^s_i - r^s_j) \tag{18}$$

Each node maintains a virtual queue for each commodity. It also maintains a *side information buffer* containing uncoded packets obtained from transmissions or overhearing.

When node i broadcasts a packet from commodity s to all receivers $j \in J$, only the receiver j^s_{iJ} puts the packet into its virtual queue corresponding to commodity s. The other nodes $J-\{j^s_{iJ}\}$ put the packet in their side information buffers (overhearing).

When node i broadcasts a coded packet from a multicast session m to all receivers $j \in J$, for each commodity $s \in m$, only the receiver j^s_{iJ} decodes the packet using overheard packets in its side information buffer and puts the packet into its virtual queue corresponding to commodity s.

E. Convergence Properties

We use $\boldsymbol{x}^\star$ and $\hat{\boldsymbol{x}}$ to denote the optimal session rate vector of problem $\mathbf{MP}$ (1) and $\mathbf{MP}-\beta$ (13) respectively. Then we can see that

$$\sum_{s\in S} U_s(\hat{x}_s) \le \sum_{s\in S} U_s(x^*_s) \le \sum_{s\in S} U_s(\hat{x}_s) + \frac{1}{\beta}\log|H| \tag{19}$$

As $\beta \to \infty$, $\hat{\boldsymbol{x}} \to \boldsymbol{x}^\star$.

Now given β, we consider the convergence properties of the overall distributed solution. Observing that there are two time scales, one is the time-scale of the primal-dual subgradient algorithm (15), the other is the underlying Markov chain.

Without the *time-scale separation assumption* on the designed Markov chain, the above primal-dual algorithm (15) turns to a stochastic primal-dual algorithm modulated by the underlying Markov chain, shown in the following:

$$\begin{cases} x_s(m+1) = \left[x_s(m) + \epsilon(m)\left(U'_s(x_s(m)) - r^s_s(m)\right)\right]_+ \\ \forall s \in S \quad \text{user rates updating} \\ r^s_i(m+1) = \left[r^s_i(m) + \epsilon(m)\left(In(m) - Out(m)\right)\right]_+ \\ \forall i \in \mathcal{N} - \{t_s\}, s \in S, \quad \text{Node prices updating} \\ r^s_{t_s}(m+1) = r^s_{t_s}(m) = 0, \forall s \in S \end{cases} \tag{20}$$

where

$$In(m) = x_s(m)\mathbb{1}_{i=s} + \sum_{j\in\mathcal{N}} \sum_{\{i|(j,I)\in\mathcal{L}, i\in I\}} (\sum_{m\in\mathcal{M}} \bar{f}^{ms}_{jIi} + \bar{f}^{v_s}_{jIi}),$$

$$Out(m) = \sum_{\{J|(i,J)\in\mathcal{L}\}} \sum_{j\in J} (\sum_{m\in\mathcal{M}} \bar{f}^{ms}_{iJj} + \bar{f}^{v_s}_{iJj}),$$

$[\cdot]_+ \triangleq \max(\cdot, 0)$, $m \in \mathbb{Z}^+$, $\mathbb{Z}^+$ denotes the set of all positive integers, $\epsilon(m)$ is the step size within the update interval T_m, $\bar{f}^{ms}_{iJj}(m)$ and $\bar{f}^{v_s}_{iJj}(m)$ are the average flow rates of session s over hyperlink *(i,J)* measured within the update interval T_m, and T_m is the time interval between the system updating $(\boldsymbol{x}(m-1), \boldsymbol{r}(m-1))$ and $(\boldsymbol{x}(m), \boldsymbol{r}(m))$.

Now we assume that $\max_{s,m} x_s(m) < \infty$, $\max_{i,s,m} r^s_i(m) < \infty$ and $U'_s(0) < \infty, \forall s \in S$. In this way, each update of stochastic prima-dual algorithm is upper bounded. For diminishing step sizes and increasing update intervals, we have the following result:

Theorem 1. *The overall solution converges asymptotically with probability one to the optimum $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{r}})$ under the following conditions:*

$$\{T_m\} \text{ is increasing with } m \tag{21}$$

$$\sum\nolimits_{m=1}^{\infty} \epsilon(m) = \infty, \ \sum\nolimits_{m=1}^{\infty} \epsilon^2(m) < \infty \tag{22}$$

$$\epsilon(m) > 0\ \forall m, \ \sum\nolimits_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} < \infty \tag{23}$$

Further, the setting $\epsilon(m) = \frac{1}{m}$, $T_m = m$, $m \ge 1$ is one specific choice satisfying conditions (21)-(23).

In practice, however, decreasing step size $\epsilon(m)$ and increasing update interval T_m beyond some certain values slow down the convergence speed considerably. So we turn to the scenario of constant step size and constant update interval. Inspired by the proof for Theorem 1, we have the following result:

Theorem 2. *The overall solution converges asymptotically with probability one to the bounded neighborhood around the optimum $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{r}})$ under the following conditions:*

$$T_m = T_0 > 0 \quad \forall m \tag{24}$$

$$\epsilon(m) = \epsilon > 0 \quad \forall m \tag{25}$$

The bounded neighborhood of $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{r}}$ is shown as follows:

$$\{(\boldsymbol{x}, \boldsymbol{r}) : |L_\beta(\boldsymbol{x}, \boldsymbol{r}) - L_\beta(\hat{\boldsymbol{x}}, \hat{\boldsymbol{r}})| \le \frac{C_3}{T_0} + \epsilon\frac{(C_1 + C_2)}{2}\},$$

where C_1, C_2, C_3 are positive constants.

III. Numerical Examples

Here we prove one numerical example: a wireless cross network shown in Fig. 1, where two unicast sessions exist: session 1 includes source A and destination E, and session 2 includes source B and destination D. We choose utility function of session 1 and 2 to be $U(\cdot) = \log(\cdot + 0.1)$ and $U(\cdot) = \log(\cdot + 0.3)$ respectively. By running the overall distributed solution with constant step size $\epsilon = 0.05$, constant update interval $T_0 = 100$ and approximating factor $\beta = 100$, we have the corresponding unicast session rates and node prices shown in Fig. 2 and Fig. 3 respectively. Focusing on unicast session rates, we compare the numerical results with theoretically optimal values in Table I. The numerical results

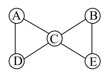

Fig. 1. A wireless cross network with three hyperlinks, (A,{C,D}), (B,{C,E}), and (C,{D,E}). All hyperlinks have one unit capacity. There are two unicast sessions, session 1 includes source A unicasting to destination E, and session 2 includes source B unicasting to destination D.

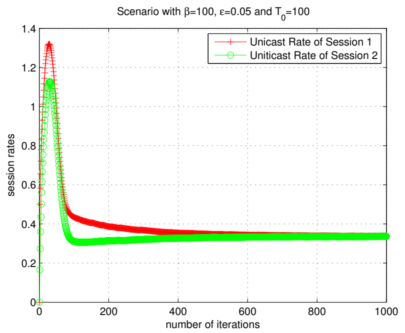

Fig. 2. Performance of the distributed solution on unicast session rates with $\beta = 100, \epsilon = 0.05, T_0 = 100$. Initial values of all session rates are *0*.

of all session rates are very close to optimums, all within 2.43% of the optimal values. More performance results can be found in our technical report [15]. These results illustrate the convergence and optimality of the proposed distributed solution.

IV. Conclusions

In this paper, we study the cross-layer optimization of wireless networks with multiple unicast sessions and network coding. We exploit the broadcast advantage of wireless medium and adopt session-decomposition based coding scheme. We provide a distributed solution including the hyperlink scheduling algorithm, primal-dual flow control algorithm, and network coding algorithm. Without the time scale separation assumption, we show the convergence to optimum with probability one under diminishing step sizes and increasing update intervals. We also show the convergence to the bounded neighborhood of optimum with probability one under constant step sizes and constant update intervals. Numerical results validate the analytical results.

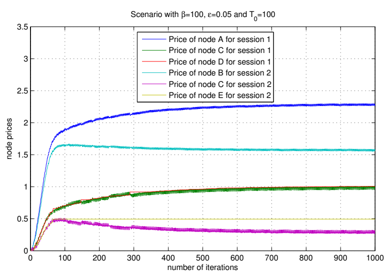

Fig. 3. Performance of the distributed solution on node prices with $\beta = 100, \epsilon = 0.05, T_0 = 100$. Initial values of all node prices are *0*. For convenience, we only show non-zero node prices. So all other node prices not shown in this picture are *0* all the time. Note that all node prices are stable.

TABLE I
PERFORMANCE COMPARISON WITH $\beta = 100, T_0 = 100, \epsilon = 0.05$

	Optimal	Approximation	Gap	Relative Error
Sum of Utility	-1.2930	-1.2785	0.0161	1.12%
Rate of Session 1	0.3333	0.3381	0.0081	2.43%
Rate of Session 2	0.3333	0.3355	0.0055	1.65%

Acknowledgment

The work described in this paper was partially supported by a grant from University Grants Committee of the Hong Kong SAR (Project No. AoE/E-02/08).

References

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
[2] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM transactions on networking*, vol. 11, no. 5, pp. 782–795, 2003.
[4] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
[5] 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*, vol. 52, no. 10, 2006.
[6] A. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *ACM-SIAM SODA*, 2004, pp. 142–150.
[7] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *ACM MobiCom*, 2007.
[8] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *IEEE INFOCOM*, 2008.
[9] A. Eryilmaz and D. Lun, "Control for inter-session network coding," in *NetCod Workshop*, 2007.
[10] T. Ho, Y. Chang, and K. Han, "On constructive network coding for multiple unicasts," in *Allerton*, 2006.
[11] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 3, 2008.
[12] A. Khreishah, C. Wang, and N. Shroff, "Optimization based rate control for communication networks with inter-session network coding," in *IEEE INFOCOM*, 2008.
[13] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in *IEEE INFOCOM*, 2007.
[14] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, "Network coding for multiple unicasts," in *IEEE ISIT*, 2006.
[15] Z. Shao and S. Y. R. Li, "Distributed Optimization for Wireless Networks with Inter-session Network Coding," *Technical Report*, 2013, available at http://home.ie.cuhk.edu.hk/~zyshao/nctp.pdf.
[16] M. Chen, S. Liew, Z. Shao, and C. Kai, "Markov Approximation for Combinatorial Network Optimization," in *IEEE INFOCOM*, 2010.