Network Utility Maximization for BATS Code Enabled Multihop Wireless Networks

Yanyan Dong, Sheng Jin, Shenghao Yang and Hoover H. F. Yin

***Abstract*—Network utility maximization (NUM) is studied for multihop wireless networks employing an efficient random linear network coding scheme called BATS codes. Compared with the classical random linear network coding scheme, BATS codes have lower computational and storage costs at the intermediate network nodes, and can achieve close-to-optimal end-to-end throughput and latency for multihop networks with packet loss. We formulate a NUM problem that optimizes the total utility of multiple communication flows under certain link scheduling constraints. Our problem employs a practical throughput measure induced by BATS codes and hence can provide realistic guidelines about network protocol designs for multihop wireless networks. Our problem in general has a non-convex objective function with integer variables, so that the algorithms of solving existing network utility maximization problems cannot be directly applied to our problem. We discuss a modified dual-based algorithm for solving our problem and evaluate its performance numerically.**

I. Introduction

Multihop wireless networks have applications including wireless ad hoc and sensor networks, vehicular networks and underwater acoustic networks. Due to the open, shared, and dynamical wireless communication media, wireless networks require more sophisticated design than the wireline networks based on dedicated and reliable links. Network coding based approach has been shown to be necessary and efficient for multihop wireless networks [1]. In this paper, we study the related network utility maximization (NUM) problem.

Optimization based approaches developed over the past twenty years provide a clean-slate framework for studying various network resource allocation problems, including congestion control and scheduling (see, for example, [2], [3]). For multihop communications, most related works in the literature assume a flow model where each network link can support certain error-free communication rate, and the concatenation of a sequence of links can support an end-to-end communication rate r if each link can support a rate r. Though this assumption is feasible when the links are error-free and the packet losses are mostly generated by congestion, the assumption does not hold in general wireless networks where link error is not avoidable due to interference, user mobility, multipath and shadowing. Therefore, research based on such a flow model cannot provide accurate guidance for designing multihop wireless networks.

Y. Dong, S. Jin and S. Yang are with The Chinese University of Hong Kong, Shenzhen, Shenzhen, China. S. Yang is also with the Shenzhen Research Institute of Big Data, Shenzhen, China. H. Yin is with the Institute of Network Coding, The Chinese University of Hong Kong, Hong Kong, China. This work was funded in part by the Shenzhen Science and Technology Innovation Committee (Grant JCYJ20170817150000267, ZDSYS20170725140921348, JCYJ20180508162604311) and NSFC (Grant 61471215).

Following the traditional store-and-forward approach for network communication, a leaky-pipe flow model has been proposed [4] for wireless networks, where only correctly received packets are forwarded to the next hop. Consider a flow of L links, where link error occurs with a probability $\epsilon \in (0,1)$. In the leaky-pipe model, the data rate decreases to $(1-\epsilon)$ fraction hop-by-hop so that the receiving rate is only $(1-\epsilon)^L$ of the rate of the source node, i.e., the end-to-end throughput decreases exponentially as the number of hops.

Theoretically, the maximum end-to-end reliable communication rate of the above network model is $1-\epsilon$, which is independent of L and can be achieved by using hop-by-hop retransmission and hop-by-hop erasure codes, e.g., fountain codes. But these approaches incur significant communication delay, computational and storage costs, and cannot be applied in practical networks, especially in networks with long latency and highly unreliable communication media. Random linear network coding (RLNC) achieves the capacity of multihop wireless networks with packet loss in a general setting [5], [6], but the classical RLNC approach is not efficient due to its high computational and storage costs, and also its high coding coefficient overhead.

BATS codes [7] were proposed to resolve these issues of RLNC. By using a small amount of computational and storage costs at the intermediate network nodes, BATS codes achieve a close-to-optimal end-to-end throughput and latency, and resolve various issues raised from the traditional hop-by-hop retransmission and hop-by-hop erasure coding approaches (see the discussion in [1]). A BATS code consists of an outer code and an inner code. As a matrix generalization of a fountain code, the outer code generates a potentially unlimited number of batches, each of which consists of a certain number (called the batch size) of coded packets. The inner code comprises linear network coding (also called recoding) at the intermediate network nodes, which is applied to the packets belonging to the same batch. (More about BATS codes in Sec II.) Using BATS codes for multihop wireless networks have been extensively discussed [8]–[12], but how to incorporate BATS codes into the NUM framework for jointly optimizing congestion control and link scheduling is largely open.

Employing BATS codes, the data flow in the network can be better understood as a *batch flow*, which has two fundamental differences compared to the traditional *packet flow*. First, as recoding is employed, there is no packet flow conservation for the intermediate network nodes. Second, the throughput of a flow cannot be simply measured by the number of packets received per unit time. Actually, the traditional packet flow can be regarded as a special case of the batch flow when the batch size is 1. The network optimization problem must be properly redesigned for this more general type of network flow.

The main purpose of this paper is to demonstrate how to properly formulate and solve the NUM problem when a BATS code is employed for a multihop wireless network. Under the network and recoding model discussed in Sec III, we consider multiple communication flows (users) in a wireless network, each of which employs a BATS code for reliable end-to-end communications. In Sec IV, we formulate a general NUM problem that optimizes the total utility of all the flows under certain link scheduling constraints. Different from the traditional counterparts, our problem uses the product of the batch rate and the expected rank of the batch transfer matrix as the measure of the end-to-end throughput of each communication flow. Our problem degenerates into a traditional one without network coding when a store-and-forward type of recoding scheme is applied, and includes some existing single flow BATS code recoding optimizations as special cases.

Our NUM problem has an objective function that may not be convex in general, and includes a sequence of integer variables for optimizing the number of recoded packets for different nodes and flows. Due to the more complicated model, the algorithms for solving the existing NUM problems cannot be directly applied to solve our problem. In Sec V, we discuss a modified dual-based algorithm, which is different from the traditional dual-based algorithms in two aspects. First, for our problem, the dual solution may not be feasible, which is usually the case in our numerical experiments, and hence a dual solution is further needed to be projected into the feasible region. Second, the integer recoding variables of a path are needed to be optimized jointly to achieve the optimal value, which is a hard problem for numerical solving. We use a heuristic algorithm to find a locally optimal solution of recoding variables. Our algorithm is evaluated numerically, and is compared with the cut-set upper bound.

II. BATS Code Basics

We briefly introduce BATS codes and some related works. Readers may find a detailed discussion of BATS codes in [1].

A. Encoding

Linear operations of this paper is applied over a base field of size q, e.g., $q = 2^8$. Let K and T be positive integers. Suppose there are K input packets for transmission, each of which is a column vector of T symbols in the base field, to be transmitted from a source node to a destination node.

A BATS code consists of an outer code and an inner code. The outer code of a BATS code generates coded packets in batches. Let M be a positive integer called the *batch size*. For $i = 1, 2, \ldots$, the ith batch $\mathbf{X}_i$ is generated from the input packets using the following steps:

1) Sample a *degree distribution* $\mathbf{\Psi} = (\Psi_1, \ldots, \Psi_K)$ which returns a degree d_i with probability Ψ_{d_i}.
2) Uniformly at random choose d_i input packets and juxtapose them to form a matrix $\mathbf{B}_i$.
3) Form a $d_i \times M$ totally random matrix $\mathbf{G}_i$, called the *batch generator matrix*.[1] Then batch $\mathbf{X}_i = \mathbf{B}_i\mathbf{G}_i$.

The generator matrices are known for decoding which can be achieved by sharing the same pseudorandom number generator at the source node and the destination node.

The inner code of a BATS code is formed by a linear network coding scheme, also called *recoding*. Recoding is applied to the coded packets belonging to the same batch so that the end-to-end transformation of each batch from the source node to the destination node is a linear operation. Let $\mathbf{H}_i$ be the *batch transfer matrix* of the ith batch and $\mathbf{Y}_i$ be the output (received) packets of the ith batch at the destination node. We have

$$\mathbf{Y}_i = \mathbf{X}_i\mathbf{H}_i = \mathbf{B}_i\mathbf{G}_i\mathbf{H}_i. \tag{1}$$

The number of columns of $\mathbf{H}_i$ corresponds to the number of packets received for the ith batch, which may vary for different batches. If no packets are received for a batch, then $\mathbf{Y}_i$ and $\mathbf{H}_i$ are empty matrices.

The batch transfer matrices of all batches can be recovered at the destination node by means of appending a coefficient vector to each of the coded packet, as in random linear network coding. Let $\mathbf{h} = (h_0, h_1, \ldots, h_M)$ be the probability distribution of the ranks of the batch transfer matrices, which is also called the *rank distribution*. The expectation of the rank distribution, also called the *expected rank*, is denoted by

$$\overline{\mathbf{h}} \triangleq \mathbb{E}[\mathbf{h}] = \sum_{i=1}^{M} i h_i.$$

The value $\overline{\mathbf{h}}$ is an upper bound on the achievable rates of the BATS code, and is achievable by the random linear outer code [13]. Moreover, the rank distribution $\mathbf{h}$ is a sufficient statistic for designing the outer code.

B. Decoding

Suppose that n batches are received at the destination node. A BATS code can be decoded by using an efficient belief propagation (BP) algorithm. The decoder knows the associated system of linear equations in (1) for $i = 1, \ldots, n$, where $\mathbf{G}_i$ and $\mathbf{H}_i$ are known. A batch with the generator matrix $\mathbf{G}$ and the batch transfer matrix $\mathbf{H}$ is said to be *decodable* if $\mathrm{rank}(\mathbf{GH})$ is equal to its degree. The BP decoding includes multiple iterations. In the first iteration, all the decodable batches are decoded by solving the associated system of linear equations, and the input packets involved in these decodable batches are recovered. In each of the following iterations, first, for each undecoded batch, all the recovered input packets involved in the batch are substituted into the associated linear system and the degree of the batch is reduced accordingly. Then, the batches becoming decodable after the updating are decoded, and the input packets involved in these decodable batches are recovered. The BP decoding stops when there is no decodable batch.

[1] A totally random matrix has uniform i.i.d. components over the base field.

In the existing theory of BATS codes, a sufficient condition was obtained such that the BP decoding can recover a given fraction of the number of input packets with high probability. A degree distribution that satisfies the sufficient condition can be obtained for a given rank distribution $\mathbf{h}$ [7]. To have all the input packets solved by BP decoding, precoding can be applied as in the Raptor codes. This theory guarantees that BATS codes can achieve a rate very close to $\overline{\mathbf{h}}$ with a low encoding/decoding computational complexity.

III. Network Model and Recoding

In this section, we first introduce a multihop wireless network model similar to the one in [2] and present the use of BATS codes for reliable end-to-end communications.

A. Network Model

We model a multihop wireless network as a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of nodes and edges respectively. An edge $(u, v) \in \mathcal{E}$ where $u, v \in \mathcal{V}$ models a communication link that transmits packets from node u to node v. Henceforth, an edge is also called a link. Associated with each link $e = (u, v)$ is a communication rate c_e packets per unit time and a packet loss rate ϵ_e, i.e., node u can transmit at most c_e packets per unit time using the link e and node v receives each packet successfully with probability $1 - \epsilon_e$ independently.

One characteristic of wireless networks is the interference between different links due to the open wireless communication media. We assume that each link e is associated with a set $\mathcal{I}_e$ of interfering links. If a link (u, v) is scheduled to transmit simultaneously with one or more interfering links, a *collision* occurs so that node v would not receive the packets transmitted by node u. Otherwise, i.e., a link (u, v) is scheduled to transmit without any interfering link which transmits simultaneously, node v would received a packet transmitted by node u with probability $1 - \epsilon_{(u,v)}$.

A *schedule* is defined by $\mathbf{s} = (s_e : e \in \mathcal{E})$ where $s_e = 1$ indicates that the link e is scheduled to transmit and $s_e = 0$ otherwise. We say a schedule $\mathbf{s}$ is *feasible* if it generates no collision, i.e., for any $e \in \mathcal{E}$, if $s_e = 1$, then $s_{e'} = 0$ for all $e' \in \mathcal{I}_e$. The collection of all feasible schedules is denoted by $\mathcal{S}$. For a feasible schedule $\mathbf{s} \in \mathcal{S}$, define a rate vector $\mathbf{r} = (c_e s_e, e \in \mathcal{E})$, where $c_e s_e$ is the maximum communication rate that link e can support when using the schedule $\mathbf{s}$. Define the *rate region* $\mathcal{R}$ by

$$\mathcal{R} = \{\mathbf{r} \colon \mathbf{s} \in \mathcal{S}\}. \tag{2}$$

The convex hull of $\mathcal{R}$ is denoted by $\mathrm{Co}(\mathcal{R})$.

We say two edges (u_1, v_1) and (u_2, v_2) are *consecutive* if $v_1 = u_2$. A sequence of links $\mathcal{P} = (e_i = (u_i, v_i) \colon i = 1, \ldots, L)$ is call a *flow* (path) of length L if e_i and e_{i+1} are consecutive for all $i = 1, \ldots, L-1$, where e_i is the ith link of the flow. In a flow $\mathcal{P}$, the first node u_1 is called the source node and the last node v_L is called the destination node.

B. Recoding Model

The intermediate network nodes recode the received packets when we use BATS codes. Here we introduce a general recoding model. We discuss the recoding scheme for a fixed flow $\mathcal{P} = \{e_i = (v_{i-1}, v_i) \colon i = 1, \ldots, L\}$. The source node v_0 uses the outer code to generate a sequence of batches of batch size M. For each batch generated by the outer code, the source node further recodes the batch to m_{e_1} packets (using certain recoding scheme to be specified later), and then transmits the recoded packets of the batch on link e_1. The recoding at the intermediate network nodes can be formulated inductively. For each batch, node v_k where $1 \le k < L$ applies random linear network coding to generate $m_{e_{k+1}}$ packets and transmit these packets on link e_{k+1}. The number of packets $m_{e_{k+1}}$ is also called the *recoding number*.

The concept of batch transfer matrices can be extended to nodes other than the destination node: for node v_i where $i = 0, \ldots, L$, the batch transfer matrix of a batch at the node v_i is formed by juxtaposing the coefficient vectors of the received packets of the batch. We call the rank of the batch transfer matrix of a batch the rank of the batch. Suppose the rank of every batch follows the same rank distribution, which can be guaranteed by our recoding schemes to be discussed later. Denote by R_i the random variable of the rank of a batch at the node v_i, where $i = 0, 1, \ldots, L$. At the source node, we have $\Pr(R_0 = M) = 1$. The probability distribution of R_L is $\mathbf{h}$. In practice, the instance of R_i, $i > 0$ can be obtained at node v_i using the packets it receives, where the coefficient vectors are embedded in the packets.

The probability distribution of R_i can be derived analytically for the following recoding schemes.

1) Random Linear Recoding: In this paper, we focus on *random linear network coding*, where each recoded packet is a linear combination of all the received packets of the batch with coefficients uniformly at random chosen from the base field. In general, $R_0, \ldots, R_L$ form a Markov chain. Denote by P_l the $(M+1) \times (M+1)$ transition matrix from R_{l-1} to R_l. As derived in [1, chap. 4], the $(i+1, j+1)$ entry of P_l is

$$P_l[i+1, j+1] = \sum_{k=j}^{m_{e_l}} \binom{m_{e_l}}{k} (1-\epsilon)^k \epsilon^{m_{e_l} - k} \zeta_j^{i,k},$$

where $\zeta_j^{i,k}$ is the probability that an $i \times k$ totally random matrix over the base field has rank j.

Denote by Y_i the random variable of the number of received packets at node v_i, where $i = 1, \ldots, L$. We know that Y_i follows the binomial distribution $\mathrm{Bi}(m_{e_i}, 1 - \epsilon_{e_i})$. As $R_i \le \min\{R_{i-1}, Y_i\} \le \min\{M, Y_1, \ldots, Y_i\}$, we have

$$\begin{aligned} \overline{\mathbf{h}} = \mathbb{E}(R_L) &\le \mathbb{E}(\min\{M, Y_1, \ldots, Y_L\}) \\ &\le \min\{M, \mathbb{E}(Y_1), \ldots, \mathbb{E}(Y_L)\} \\ &= \min\{M, m_{e_i}(1 - \epsilon_{e_i}) \colon i = 1, \ldots, L\}, \end{aligned} \tag{3}$$

which is the cut-set upper bound of the expected rank.

2) Store-and-Forward: We can define a special recoding scheme to emulate the behavior of store-and-forward. The recoding scheme is as follows: for node v_k where $k = 0, 1, \dots, L-1$,

- If $m_{e_{k+1}} \le R_k$, then node v_k transmits $m_{e_{k+1}}$ packets among the ones it received with linearly independent coefficient vectors;
- If $m_{e_{k+1}} > R_k$, then node v_k transmits R_k received packets with linearly independent coefficient vectors, together with $m_{e_{k+1}} - R_k$ all-zero packets.

Therefore, under the condition that $\min\{R_k, m_{e_{k+1}}\} = t_{k+1}$, Y_{k+1} follows the binomial distribution $\mathrm{Bi}(t_{k+1}, 1-\epsilon_{e_{k+1}})$. Consider two special cases:

- When $M = m_{e_1} = \cdots = m_{e_L} = 1$, then $\Pr(R_k = 1) = \prod_{i=1}^{k}(1-\epsilon_{e_i})$ for $k = 1, \dots, L$.
- When $\epsilon_{e_i} = 0$ for all i, then $R_k = \min\{M, m_{e_i} \colon i = 1, \dots, k\}$.

IV. Network Utility Maximization

We formulate the network optimization problem for multi-hop wireless networks with BATS codes employed and discuss some special cases of the optimization problem.

A. Utility Maximization

We consider a fixed number of communication flows in the network, where the ith flow is denoted by a path $\mathcal{P}^i$. Each flow employs BATS codes independently. For the ith flow $\mathcal{P}^i$, the source node generates batches of batch size M^i and transmits α_i batches per unit time. For all the batches transmitted, network nodes in $\mathcal{P}^i$ employ certain recoding schemes as we described in Sec III-B.

For an edge $e \in \mathcal{E}$, if e is used by the flow $\mathcal{P}^i$, we denote by m_e^i the number of recoded packets transmitted on link e for a batch in flow $\mathcal{P}^i$. Let $\mathbf{h}^i$ be the rank distribution of the end-to-end batch transfer matrix for the ith flow. The throughput of the ith flow is $\alpha_i \overline{\mathbf{h}^i} = \alpha_i \mathbb{E}[\mathbf{h}^i]$, which can be achieved by BATS codes (see Sec. II).

We can choose the parameter $\{\alpha_i, \mathbf{m}^i = (m_e^i \colon e \in \mathcal{P}^i)\}$ to jointly optimize all the flows. Motivated by traditional network utility maximization researches, such a joint optimization problem can be formulated as follows:

$$
\begin{aligned}
\max_{\{\alpha_i, \mathbf{m}^i\}, \mathbf{r}} \quad & \sum_i U_i(\alpha_i \overline{\mathbf{h}^i}) \\
\text{s.t.} \quad & \sum_{i: e \in \mathcal{P}^i} \alpha_i m_e^i \le r_e, \quad \forall e \in \mathcal{E} \\
& \mathbf{r} \in \mathrm{Co}(\mathcal{R}),
\end{aligned} \tag{4}
$$

where each U_i is a certain non-decreasing concave utility function. In the above optimization problem, $m_e^i \ge 0$ takes integer values and $\alpha_i \ge 0$ is real. If link e is used for the ith flow, $\alpha_i m_e^i$ packets are transmitted on e by the flow per unit time. The constraints in the above optimization problem bound the communication rate for each link.

B. Traditional Cases: Store-and-Forward Recoding

This problem includes the classical network utility maximization as special cases when applying store-and-forward:

1) No Loss: In the case that the packet loss rate $\epsilon_e = 0$ for all links e and store-and-forward is applied (see Sec. III-B2), we have $\overline{\mathbf{h}^i} = \min\{M^i, m_e^i \colon e \in \mathcal{P}^i\}$. By letting $f_e^i = \alpha_i m_e^i$, the optimization problem (4) becomes

$$
\begin{aligned}
\max_{\{f_e^i\}, \mathbf{r}} \quad & \sum_i U_i(\min\{f_e^i \colon e \in \mathcal{P}^i\}) \\
\text{s.t.} \quad & \sum_{i: e \in \mathcal{P}^i} f_e^i \le r_e, \quad \forall e \in \mathcal{E} \\
& \mathbf{r} \in \mathrm{Co}(\mathcal{R}),
\end{aligned} \tag{5}
$$

where we use the fact that $\alpha_i M^i$ is unconstrained and hence is omitted from $\min\{\alpha_i M^i, f_e^i \colon e \in \mathcal{P}^i\}$. This is a traditional network utility maximization problem (see [2], [3]).

2) Leaky Pipe: When store-and-forward is applied with $M^i = m_{e_1}^i = \cdots = m_{e_L}^i = 1$ (see Sec. III-B2), our problem (4) becomes one of the leaky pipe model studied in [4].

C. Special Interference Models

When applying random linear recoding, our problem (4) in general has a more complicated objective function than the traditional ones discussed above. We can let $f_e^i = \alpha_i m_e^i$ to linearize the constraints and use $\lfloor f_e^i / \alpha_i \rfloor$ to replace m_e^i in $\mathbf{h}^i$ to remove the discrete variables. But the expected rank $\mathbf{h}^i$ is not necessary convex nor continuous in terms of f_e^i and α_i. These facts make optimization problem (4) usually more difficult to solve than the traditional counterpart.

For BATS codes, couple special cases of optimization problem (4) with only one flow have been studied in literature. When there is only one flow $\mathcal{P}$, (4) becomes

$$
\begin{aligned}
\max_{\alpha, \{m_e \colon e \in \mathcal{P}\}, \mathbf{r}} \quad & \alpha \overline{\mathbf{h}} \\
\text{s.t.} \quad & \alpha m_e \le r_e, \quad \forall e \in \mathcal{P} \\
& \mathbf{r} \in \mathrm{Co}(\mathcal{R}),
\end{aligned} \tag{6}
$$

where we omit the utility functions as they are non-decreasing.

1) Single flow, no collision: The recoding optimizations in [1] focus on the case that links have the same rate c and no collision occurs so that $\mathcal{R} = \{(c, \dots, c)\}$. This rate region is suitable for multi-ratio multi-channel wireless networks. As $\overline{\mathbf{h}}$ is a non-decreasing function of m_e, with this rate region, the optimization problem (6) becomes

$$
\max_{\{m_e = M' \colon e \in \mathcal{E}\}} \overline{\mathbf{h}}/M',
$$

which can be solved easily by exploring a range of integer values of M'.

2) Single flow, all collision: The collision model in [12] is that, only one link can transmit at the same time, and hence $\mathcal{R} = \{(\alpha_e, e \in \mathcal{E}) \colon \sum_e \alpha_e = c\}$. This rate region is suitable for the case that all nodes are very close to each other. With this rate region, the optimization problem (6) becomes

$$
\max_{\{m_e \colon e \in \mathcal{E}\}} \overline{\mathbf{h}}/\textstyle\sum_e m_e.
$$

Several algorithms for solving the above optimization problem have been discussed in [12].

In this paper, we will formulate the problem for a general collision model and multiple flows.

D. Cut-set Upper Bound

We first derive an upper bound for the optimization problem (4) with the random linear recoding discussed in Sec. III-B1. By the upper bound shown in (3), we have

$$\overline{\mathbf{h}^i} \leq \min\{M^i, m_e^i(1-\epsilon_e) \colon e \in \mathcal{P}\}.$$

By letting $f_e^i = \alpha_i m_e^i(1-\epsilon_e)$ and replacing $\overline{\mathbf{h}^i}$ by the above upper bound, the optimization problem (4) becomes

$$\begin{aligned} \max_{\{f_e^i\},\mathbf{r}} \quad & \sum_i U_i(\min\{f_e^i \colon e \in \mathcal{P}^i\}) \\ \text{s.t.} \quad & \sum_{i\colon e\in\mathcal{P}^i} f_e^i \leq r_e(1-\epsilon_e), \quad \forall e \in \mathcal{P} \\ & \mathbf{r} \in \mathrm{Co}(\mathcal{R}). \end{aligned} \tag{7}$$

The optimal value of (7) is an upper bound on the optimal value of (4). This is also called the cut-set bound of the network communication capacity. As a traditional network utility maximization problem, (7) can be solved by using, e.g., the algorithms introduced in [2], [3]. This upper bound, however, is not achievable in practical scenarios where latency or limited storage and computational power are considered.

V. Utility Maximization Algorithm

The traditional network utility maximization problems can be solved by using primal, dual or primal-dual approaches [3]. As the problem decomposition induced by the dual approach has the advantage of simplifying the complexity [2], we adopt this approach for our optimization problem. However, due to the more complex model, it is not straightforward to solve our problem using the traditional dual approach. First, the subproblem (induced by the dual approach) optimizing the batch rates and recoding numbers is hard to solve as the integer recoding parameters cannot be decoupled. Second, the dual solution is not feasible as the objective function is not convex in general.

A. Dual-Based Algorithm

Associating a Lagrange multiplier q_e for each inequality constraint in (4), the Lagrangian is

$$\begin{aligned} & \sum_i U_i(\alpha_i\overline{\mathbf{h}^i}) - \sum_{e\in\mathcal{E}} q_e \left(\sum_{i:e\in\mathcal{P}^i} \alpha_i m_e^i - r_e\right) \\ = & \sum_i \left[U_i(\alpha_i\overline{\mathbf{h}^i}) - \alpha_i \sum_{e\in\mathcal{P}^i} q_e m_e^i\right] + \sum_{e\in\mathcal{E}} q_e r_e, \end{aligned}$$

where q_e is an implicit cost for the link e. The dual problem of (4) is to minimize

$$\sum_i \max_{\alpha_i,\mathbf{m}^i} \left[U_i(\alpha_i\overline{\mathbf{h}^i}) - \alpha_i \sum_{e\in\mathcal{P}^i} q_e m_e^i\right] + \max_{\mathbf{r}\in\mathrm{Co}(\mathcal{R})} \sum_{e\in\mathcal{E}} q_e r_e.$$

Similar to [2], we have the following solution of the dual problem. The batch rate $\alpha_i(t)$ and the recoding number $m_e^i(t)$ are updated by solving

$$(\alpha_i(t), \mathbf{m}^i(t)) = \arg\max_{\alpha_i,\mathbf{m}^i} \left[U_i(\alpha_i\overline{\mathbf{h}^i}) - \alpha_i \sum_{e\in\mathcal{P}^i} q_e(t) m_e^i\right] \tag{8}$$

for each flow i. The schedule is determined by

$$\mathbf{r}(t) = \arg\max_{\mathbf{r}\in\mathrm{Co}(\mathcal{R})} \sum_{e\in\mathcal{E}} q_e(t) r_e. \tag{9}$$

The Lagrange multipliers are updated by

$$q_e(t+1) = \left[q_e(t) + \gamma_t \left(\sum_{i:e\in\mathcal{P}^i} \alpha_i(t) m_e^i(t) - r_e(t)\right)\right]^+ \tag{10}$$

where $[x]^+ = \max\{0, x\}$ for real x, and γ_t, $t = 1, 2, \ldots$ is a sequence of positive step sizes such that the subgradient search converges, e.g., $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$.

In the above algorithm, the optimization of the batch rates and recoding numbers for each flow is decoupled, so that the complexity is linear with the number of flows. Moreover, the scheduling update in (9) is the same as the one in the traditional problem (see [2, eq. (12)]), so that the same imperfect scheduling policies discussed in [2] can be applied to simplify the complexity of finding an optimal scheduling. However, our problem is more complicated than the traditional ones in the following two aspects.

First, the subproblem (8) optimizes the recoding numbers alone a path jointly. As the integer recoding numbers get involved in the complicated formula of $\overline{\mathbf{h}^i}$, experiments show that it is not optimal to alternately optimizes one recoding number while fixing the others. The exhaustive search of the optimal recoding numbers has an exponential complexity in terms of the path length. To reduce the complexity, we limit each recoding number to a small neighboring range in each iteration to find a local optimal solution.

Second, as the primal (4) is non-concave, the optimizer of the dual problem may not be feasible for the primal one. Therefore, after obtaining a dual solution $\{(\tilde{\alpha}_i, \tilde{\mathbf{m}}^i)\}$ by multiple rounds of updating using (8)-(10), we find a feasible primal solution by solving

$$\begin{aligned} \max_{\alpha_i,\mathbf{r}} \quad & \sum_i U_i(\alpha_i\overline{\mathbf{h}^i}(\tilde{\mathbf{m}}^i)) \\ \text{s.t.} \quad & \sum_{i:e\in\mathcal{P}^i} \alpha_i \tilde{m}_e^i \leq r_e, \quad \forall e \in \mathcal{E} \\ & \mathbf{r} \in \mathrm{Co}(\mathcal{R}), \end{aligned}$$

where we fix the recoding numbers of the dual solution and scale the batch rates to make the solution feasible.

B. Numerical Results

For (4) with the natural logarithm as the utility functions (same as TCP Vegas [14]), we evaluate the above dual-based algorithm using a network with node set $\mathcal{V} = \{v_0, \ldots, v_8\}$ and edge set $\mathcal{E} = \{e_i = (v_{i-1}, v_i), i = 1, \ldots, 8\}$. Let $\epsilon_i = \epsilon_{e_i}$ and $c_i = c_{e_i}$. We test 11 cases of different flows, link loss rates and communication rates as listed in Table I. For BATS codes, we use batch size $M = 16$ and base field size $q = 256$ for all the flows. We use the two-hop interference model.

The numerical results of solving these 11 cases are given in Tables II and III. Tables II includes the recoding numbers obtained for each flow. We see from the "utility" column of Table III that the utilities of the two flows $U_1 = \log(\alpha_1\overline{\mathbf{h}^1})$

TABLE I
NETWORK AND FLOW SETTINGS

case	flows	loss rate	commun. rate
1	$[11111000; 00111111]$	/	/
2	same as case 1	/	$c_{3,4,5} = 2$
3	same as case 1	/	$c_{1,2,6,7,8} = 1/2$
4	same as case 1	/	$c_{1,2,6,7,8} = 1/4$
5	same as case 1	$\epsilon_{3,4,5} = 0.1$	/
6	same as case 1	$\epsilon_{3,7} = 0.1$	/
7	same as case 1	$\epsilon_{1,2,6,7,8} = 0.1$	/
8	same as case 1	$\epsilon_{1,2,6,7,8} = 0.4$	/
9	$[11111111; 11111111]$	/	/
10	$[11111111; 00111111]$	/	/
11	$[11111111; 00111100]$	/	/

[a] In the "flows" column, $[11111000; 00111111]$ means that the first flow contains the links e_1 to e_5 and the second flow contains the links e_3 to e_8.
[b] In the "loss rate" and "commun. rate" columns, without otherwise specified, $\epsilon_i = 0.2$ and $c_i = 1$. Here $\epsilon_{3,4,5} = 0.1$ means $\epsilon_3 = \epsilon_4 = \epsilon_5 = 0.1$, and $c_{3,4,5} = 2$ means $c_3 = c_4 = c_5 = 2$.

TABLE II
RECODING NUMBERS OBTAINED BY THE DUAL-BASED ALGORITHM

case	$\mathbf{m}^1$	$\mathbf{m}^2$
1	$[29, 30, 19, 19, 19]$	$[19, 19, 19, 30, 29, 30]$
2	$[21, 20, 20, 21, 21]$	$[20, 21, 21, 20, 21, 21]$
3	$[20, 20, 20, 21, 21]$	$[20, 21, 21, 20, 21, 22]$
4	$[19, 19, 22, 25, 24]$	$[22, 26, 24, 20, 20, 20]$
5	$[29, 30, 19, 19, 19]$	$[19, 19, 19, 30, 29, 30]$
6	$[29, 30, 17, 19, 19]$	$[17, 19, 19, 30, 24, 30]$
7	$[24, 24, 19, 19, 19]$	$[19, 19, 19, 24, 24, 24]$
8	$[35, 34, 18, 18, 18]$	$[18, 18, 18, 35, 34, 35]$
9	$[22, 22, 21, 22, 22, 21, 22, 22]$	$[22, 22, 21, 22, 22, 21, 22, 22]$
10	$[30, 30, 20, 20, 20, 20, 20, 20]$	$[21, 21, 21, 21, 21, 21]$
11	$[30, 30, 20, 19, 19, 20, 30, 30]$	$[21, 20, 20, 21]$

and $U_2 = \log(\alpha_2 \overline{\mathbf{h}^2})$ are very close, which means that the flow control induced by the algorithm is fair.

For each case, we also compare the total utility of the two flows $U = U_1 + U_2$ with the optimal value $\tilde{U}$ of (7) which is the cut-set bound. We give the value

$$\kappa = \exp((U - \tilde{U})/2)$$

in the last column of Table III. To see the role of κ, we scale α_i by $1/\kappa$ and then evaluate the total utility:

$$\log(\alpha_1 \overline{\mathbf{h}^1}/\kappa) + \log(\alpha_2 \overline{\mathbf{h}^2}/\kappa) = -2\log(\kappa) + U = \tilde{U}.$$

In other words, κ is the rate scaling factor so that the solution can achieve the cut-set bound. We know that $\kappa \le 1$ and observe that κ is at least 77.65% for our evaluated cases. For the "cut-set bound" column in Table III, we see that the cut-set bounds are the same for 6 cases. The reason is that all these cases share the same "bottleneck" on links e_3, e_4 and e_5.

TABLE III
COMPARISON TABLE

case	$[\alpha_1, \alpha_2]/100$	utility $[U_1, U_2]$	cut-set bound	κ
1	$[0.877, 0.877]$	$[-2.119, -2.119]$	-4.030	90.12%
2	$[1.620, 1.610]$	$[-1.470, -1.484]$	-2.644	85.64%
3	$[0.823, 0.790]$	$[-2.159, -2.187]$	-4.030	85.38%
4	$[0.535, 0.423]$	$[-2.597, -2.836]$	-5.215	89.67%
5	$[0.877, 0.877]$	$[-1.992, -2.089]$	-3.794	86.63%
6	$[0.909, 0.909]$	$[-2.071, -2.119]$	-3.954	88.87%
7	$[0.877, 0.877]$	$[-2.119, -2.052]$	-4.030	93.19%
8	$[1.000, 0.875]$	$[-2.090, -2.446]$	-4.030	77.65%
9	$[0.769, 0.769]$	$[-2.191, -2.191]$	-4.030	83.86%
10	$[0.833, 0.794]$	$[-2.173, -2.172]$	-4.030	85.43%
11	$[0.862, 0.820]$	$[-2.137, -2.138]$	-4.030	88.47%

VI. CONCLUDING REMARKS

Towards a large-scale deployment of BATS code based network communication protocols, it is necessary to study network resource allocation problems. In this paper, we demonstrated how to incorporate BATS codes naturally into a network utility maximization framework for congestion control and scheduling. Our problem includes the traditional ones without network coding as special cases, but is in general more complicated to solve due to the non-concave objective function. In this paper, we showed a dual-based algorithm for centralized solving the problem. For future works, we need to study how to solve the recoding number subproblem more efficiently, and we should study distributed algorithms that can be implemented in real systems.

REFERENCES

[1] S. Yang and R. W. Yeung, *BATS Codes: Theory and Practice*, ser. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2017.

[2] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, 2006.

[3] S. Shakkottai and R. Srikant, "Network optimization and control," *Foundations and Trends® in Networking*, vol. 2, no. 3, pp. 271–379, 2008.

[4] Q. Gao, J. Zhang, and S. V. Hanly, "Cross-layer rate control in wireless networks with lossy links: leaky-pipe flow, effective network utility maximization and hop-by-hop algorithms," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 3068–3076, 2009.

[5] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 789–804, 2006.

[6] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, 2008.

[7] S. Yang and R. W. Yeung, "Batched sparse codes," *IEEE Trans. Inform. Theory*, vol. 60, no. 9, pp. 5322–5346, Sep. 2014.

[8] Q. Huang, K. Sun, X. Li, and D. O. Wu, "Just FUN: A joint fountain coding and network coding approach to loss-tolerant information spreading," in *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2014, pp. 83–92.

[9] S. Yang, R. W. Yeung, H. F. Cheung, and H. H. Yin, "BATS: Network coding in action," in *2014 52nd Annual Allerton Conf. Comm., Control, and Computing*, Oct. 2014, pp. 1204–1211.

[10] X. Xu, M. S. G. P. Kumar, Y. L. Guan, and P. H. J. Chong, "Two-phase cooperative broadcasting based on batched network code," *IEEE Trans. Comm.*, vol. 64, no. 2, pp. 706–714, Feb. 2016.

[11] S. Yang, J. Ma, and X. Huang, "Multi-hop underwater acoustic networks based on BATS codes," in *Proceedings of the Thirteenth ACM International Conference on Underwater Networks & Systems*, 2018, pp. 30:1–30:5.

[12] Z. Zhou, C. Li, S. Yang, and X. Guang, "Practical inner codes for BATS codes in multi-hop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2751–2762, Mar. 2019.

[13] S. Yang, J. Meng, and E.-h. Yang, "Coding for linear operator channels over finite fields," in *Proc. IEEE ISIT '10*, Jun. 2010, pp. 2413–2417.

[14] S. H. Low, *Analytical methods for network congestion control*, ser. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2017.