2012 International Symposium on Network Coding (NetCod)

Opportunistic Network Coding: Competitive Analysis

Yu-Pin Hsu and Alex Sprintson
Department of Electrical and Computer Engineering
Texas A&M University
{yupinhsu, spalex}@tamu.edu

***Abstract*—It was recently shown that the *reverse carpooling* technique can realize the benefits of network coding while requiring only a simple coding scheme that can be efficiently implemented in practice. However, when no opportunity for coding exists at an intermediate (relay) node, it needs to decide whether to transmit an uncoded packet or wait for the next opportunity to create a coded packet. While the decision to wait for the next coding opportunity can reduce the number of transmissions, it incurs a penalty in terms of packet delay. In this paper we present an on-line algorithm for making transmit/wait decisions at the relay nodes. The algorithm minimizes the total system cost that includes both the number of transmissions as well as packet delay. Our algorithm is based on the primal-dual approach and achieves the competitive ratio of $e/(e-1)$.**

I. Introduction

In recent years, there has been a growing interest in the applications of network coding techniques in multi-hop wireless networks. It was shown that network coding can significantly minimize the number of transmissions resulting in energy savings and, in turn, longer battery life. The *reverse carpooling* technique allows to realize the benefits of network coding while requiring only simple encoding and decoding algorithms [1].

The reverse carpooling technique for a simple relay network is demonstrated in Fig. 1(a). In this example, nodes n_1 and n_2 exchange packets through a relay node R. The traditional approach that does not involve network coding requires four transmissions (two for each packet). In contrast, the network coding approach only requires three transmissions. In particular, with the network coding approach nodes n_1 and n_2 send packets x_1 and x_2 to the relay node, and the relay node broadcasts the linear combination $x_1 + x_2$ (e.g., over field $GF(2^n)$) to both n_1 and n_2. Each of the nodes n_1 and n_2 can then obtain the packet it needs by subtracting the packet it transmitted previously from the mixed packet $x_1 + x_2$.

In a more general case, reverse carpooling includes two flows that traverse a path in opposite directions. For example, Fig. 1(b) shows two flows, from n_1 to n_4 and from n_4 to n_1 that share a common path (n_1, n_2, n_3, n_4) with two relay nodes n_2 and n_3. It is easy to verify that the network coding approach can save up to 50% of transmissions. Effectively, once one connection has been established, another connection in the opposite direction can be supported with small additional cost.

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0954153.

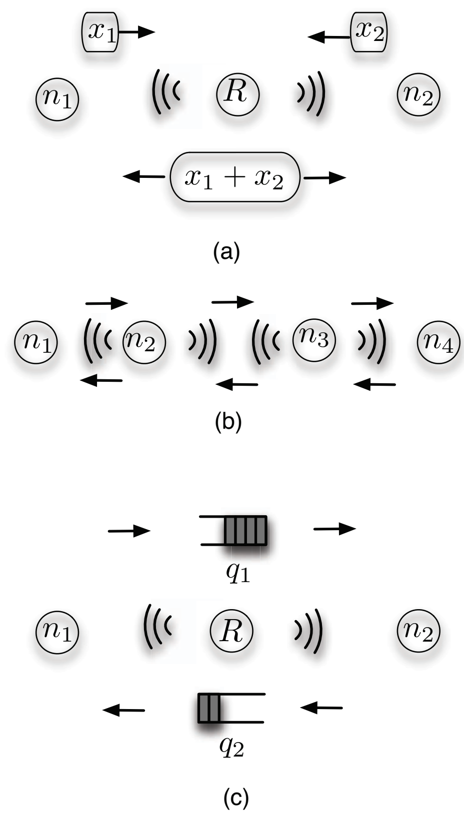

Fig. 1. (a) Wireless Network Coding (b) Reverse carpooling (c) 3-Node Relay Network.

To achieve the maximum benefit from the network coding, we need to make sure that the relay nodes can create a sufficient number of coded packets. Note that a relay node can only create a coded packet when it has at least one packet to transmit in each direction. More specifically, consider relay node R depicted in Fig. 1(c). Relay R maintains two queues, q_1 and q_2 that store packets that need to be delivered to nodes n_2 and n_1, respectively. If both queues are not empty, then R can construct a coded packet by combining the packets from the top of both queues. However, what should the relay do if there are packets waiting to be transmitted in one of the queues, while the second queue is empty? Should the relay wait for a coding opportunity or just transmit a packet from a non-empty queue without coding? Waiting for a coding opportunity would reduce the number of transmissions, but incur a certain penalty in terms of delay. Transmitting an uncoded packet would minimize the delay, but would result in a larger number of transmissions.

In this paper we present an on-line algorithm for this problem that strikes a trade-off between the average delay and the number of transmissions. We analyze the performance of this algorithm using the *competitive analysis* [2] techniques which characterize the performance of the algorithm in the worst-case scenario.

A. Related Work

Network coding research was initiated by the seminal work [3] and since then attracted a significant interest from the research community. Wireless network coding has been considered in many recent studies, e.g., [4]. In particular, reference [4] proposes an architecture called COPE, which contains a special network coding layer between the IP and MAC layers. In [5], an opportunistic routing protocol is proposed, referred to as MORE, that randomly mixes packets that belong to the same flow before forwarding them to the next hop. Several works, e.g., [6–11], investigate the scheduling or/and routing problems in the coding networks. However, in contrast to previous works mentioned above, we assume that the client can wait for a coding opportunity. In our earlier work [12], we applied the Markov decision process techniques to study a similar problem where the arrival process is governed by a certain probability distribution. The purpose of [12] is to identify the structure of the optimal strategy and to describe a procedure to find it.

B. Our contribution

We develop an on-line algorithm that makes a decision on whether to transmit or to wait at each time instant. The objective of the algorithm is to minimize the sum of the transmission and waiting time costs. Thus, at any time slot, the decision of transmitting a packet would incur a certain transmission cost, while the decision to wait would incur a waiting cost, but will possibly allow to share the transmission cost between two packets in the future.

In this paper we focus on the setting with a single relay node. In contrary to our previous work [12] where the statistical characteristics of the arrival process are assumed to be known, in this work we analyze the problem from the viewpoint of competitive analysis. We propose an on-line algorithm for the relay that makes wait/transmit decisions without the information of the future arrivals.

We can think of an on-line problem as a game between a decision maker and an adversary. The decision maker wants to get better performance in the presence of uncertainty. However, the malicious adversary attempts to construct the worst possible arrival process called *adversarial arrival process*. The performance of the on-line algorithm is analyzed using the *competitive analysis techniques* [13]. The proposed algorithm can achieve the competitive ratio of $e/(e-1)$.

II. System Overview

We consider a multi-hop wireless network operating a time-division multiplexing scheme to forward packets from various sources to destinations. Time is divided into small intervals, referred to as *slots*. Each slot is divided, in turn, into several *mini-slots*, one for each node in the system, including the relay. The packets can be transmitted during each mini-slot by a node assigned to that mini-slot.

A. Scenario from a Relay's Perspective

The basic scenario considered in the paper is depicted on Fig. 1(c). We assume that there is a packet flow f_1 that goes from node n_1 to n_2 and another flow f_2 from node n_2 to n_1, both of which are through relay R. The packets from both flows are stored at separate queues, q_1 and q_2, at node R. We assume that both queues are FIFO (first-in first-out). In this scenario, each time slot is divided into three mini-slots, such nodes n_1 and n_2 are assigned to the first two mini-slots and the relay node is assigned to the third mini-slot. In this paper we assume that there is no upper bound on the number of packets that can be transmitted by the relay during its mini-slot. However, our algorithm will not require more than four packets that will be transmitted by the relay during one mini-slot.

Note that the relay gets an opportunity to transmit at the end of the slot. When both queues are non-empty, the relay pulls one packet from q_1 and one from q_2 and transmits a linear combination of the packets (e.g., their sum over $GF(2^n)$). Note that in this case, the relay is able to take the advantage of the network coding, reducing the number of transmitted packets by one. We refer to this case a *coding opportunity*. Note that if both of the queues are empty, that the relay will not able to transmit during its mini-slot. In the case when one of the queues is empty and the second one is not, the relay has to make a decision between the following two options:

1) transmit a packet from an non-empty queue without coding;
2) wait for the next slot in the hope to receive a matching packet in the other queue and be able to utilize the coding opportunity.

Note that the first option increases the number of transmissions, while the second option increases packet delay in the hope of reducing the number of transmissions.

B. Competitive analysis

We develop an on-line algorithm for the decision problem at the relay and analyze it using the competitive analysis techniques. Note that relay has to make its decisions without the knowledge of the future arrivals. For $i \in \{1,2\}$ and $t = 0,1,2,\cdots$, let $Q_t^{(i)}$ be the number of packets stored at queue q_i at slot t, just before the beginning of the relay's mini-slot. We denote by $Q_t = (Q_t^{(1)}, Q_t^{(2)})$ the state of the system. We also introduce the indicator $D_t \in \{0,1\}$ such that

$D_t = 0$ if the decision at time slot t was to wait and $D_t = 1$ if the decision at time slot t was to transmit. As discussed above, if $Q_t^{(1)} = Q_t^{(2)} = 0$, then $D_t = 0$. Also, if $Q_t^{(1)} > 0$ and $Q_t^{(2)} > 0$, then $D_t = 1$. When exactly one of $Q_t^{(1)}$ and $Q_t^{(2)}$ is non-zero, then the on-line algorithm has to make a decision on whether to transmit an uncoded packet or to wait.

We use $A_t^{(i)} \in \{0, 1\}$ to indicate an arrival of the packet to q_i at slot t, such that $A_t^{(i)} = 0$ if no packet arrived at slot t and $A_t^{(i)} = 1$ otherwise. The *arrival pattern* $I = \{(A_t^{(1)}, A_t^{(2)})\}_{t=0}^{\infty}$ captures the arrival of packets to both queues. We assume that there will be a finite number of packet arrivals to both queues.

The number of packets in at slot $t+1$ can be expressed as follows:

$$Q_{t+1}^{(i)} = [Q_t^{(i)} - D_t]^+ + A_{t+1}^{(i)}, \tag{1}$$

where $[x]^+ = \max(x, 0)$.

We proceed to define transmission and delay costs. We define by C_T the cost of transmitting a packet and C_H be the cost for holding a packet for one slot. We assume that if a packet is transmitted in the same slot it arrived, its delay is zero. Clearly, the cost of transmitting a coded packet is the same as that of an uncoded packet.

Let $C(Q_t, D_t)$ be the *immediate cost* incurred at time t if action D_t is taken when the system is in state Q_t:

$$C(Q_t, D_t) = C_H([Q_t^{(1)} - D_t]^+ + [Q_t^{(2)} - D_t]^+) + C_T D_t. \tag{2}$$

Without loss of generality, we assume that $C_H = 1$.

We define an *algorithm* $\theta = \{D_0, D_1, \dots\}$ as a sequence of decisions made at different time slots. An algorithm θ can be deterministic or randomized, where a randomized algorithm is a distribution over deterministic algorithms. Note that with a randomized algorithm, $\{D_t | t = 0, 1, \cdots\}$ are random variables. For the given arrival pattern $I = \{(A_t^{(1)}, A_t^{(2)})\}_{t=0}^{\infty}$ to q_1 and q_2, the total cost $V(I, \theta)$ of a deterministic algorithm θ at time T is defined as:

$$V(I, \theta) = \sum_{t=0}^{\infty} C(Q_t, D_t). \tag{3}$$

For a randomized algorithm, we define the *expected cost* $V(I, \theta)$ of θ as follows:

$$V(I, \theta) = \mathbb{E}_\theta \left[\sum_{t=0}^{\infty} C(Q_t, D_t) \right], \tag{4}$$

where the expectation is over the randomized policy θ.

An *off-line* algorithm knows the arrival pattern in advance. We denote by $OPT(I)$ the cost of an optimal off-line algorithm, i.e., algorithm that minimizes $V(I, \theta)$. In contrast to off-line algorithms, an *on-line* algorithm makes decisions at time t without knowing $A_j^{(i)}, i \in \{1, 2\}$ for $j > t$.

An on-line algorithm θ is said to be c-competitive if for every input pattern I it holds that

$$V(I, \theta) \le c \cdot OPT(I) + \alpha, \tag{5}$$

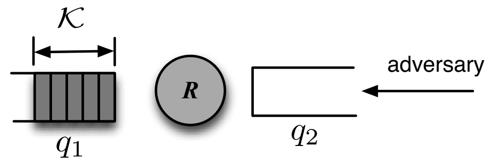

Fig. 2. One adversarial process to q_1.

where α is a constant independent of I.

In this paper we focus on the following setting:

- q_1 has $\mathcal{K}$ packets at time 0 (i.e., $Q_0^{(1)} = \mathcal{K}$);
- there are no arrivals to q_1 (i.e., $A_t^{(1)} = 0$ for all t);
- q_2 has zero packets at time 0, (i.e., $Q_0^{(2)} = 0$).
- the adversary controls the arrival process of exactly $\mathcal{K}$ packets to q_2.

Fig. 2 illustrates the scenario considered in this paper.

We assume that $\mathcal{K} \le C_T$; otherwise, there is no potential benefit for the relay to wait for a coding opportunity.

In the rest of the paper we propose an on-line algorithm for this problem that achieves a competitive ratio of $e/(e-1)$.

III. Primal-dual formulation

We need to introduce some additional notation. Consider a deterministic algorithm θ. First, let x be the number of packets in q_1 transmitted by the algorithm without coding. We also denote by z_t the number of packets waiting in q_1 at time t and by $\eta_t = \sum_{\tau=0}^{t} A_\tau^{(2)}$ we define the number of packets added to q_2 by the adversary by time t.

We observe that without loss of generality, we can assume that the algorithm will never delay packets that arrive to q_2. Indeed, since there are no arrivals to q_1 delaying a packet in q_2 only contributes to the increase in the overall waiting time.

The cost $C(I, \theta)$ of θ can be expressed as:

$$\sum_{t=0}^{\infty} C(Q_t, D_t) = C_T \cdot \mathcal{K} + C_T \cdot x + \sum_{t=0}^{\infty} z_t. \tag{6}$$

Note that the first term in Equation (6) covers the cost of transmitting all packets in q_2 that were sent uncoded, as well as the cost of coded packets. The second term, $C_T \cdot x$ covers the cost of transmitting uncoded packets from q_1. Finally, the term $\sum_{t=0}^{\infty} z_t$ covers the cost associated with waiting time of packets in q_1.

The off-line decision problem can be formulated as the following integer program:

Primal program:

$$\min \quad C_T \cdot x + \sum_{t=0}^{\infty} z_t \tag{7}$$

$$\text{s.t.} \quad x + z_t \ge \mathcal{K} - \eta_t \quad \forall t \ge 0 \tag{8}$$

$$x, z_t \in \{0, 1, 2, \cdots\} \ \forall t \ge 0 \tag{9}$$

Note that the objective function (7) is identical to (6), excluding the constant term $C_T \cdot \mathcal{K}$. Constraint (8) specifies that the number of packets stored at q_1 at time slot t is at least $\mathcal{K} - \eta_t - x$. Indeed, out of $\mathcal{K}$ packets that were at queue q_1 in the beginning of the algorithm, at most x could be sent by time t without coding and at most η_t could be sent as a combination of packets from q_2.

The integer formulation can be relaxed by substituting the integrality constraint (9) with a non-negativity constraint $x, z_t > 0$ for all $t \geq 0$. The dual program of the resulting linear program can be formulated as follows:

Dual program:

$$\max \quad \sum_{t=0}^{\infty} (\mathcal{K} - \eta_t) y_t \tag{10}$$

$$\text{s.t.} \quad \sum_{t=0}^{\infty} y_t \leq C_T \tag{11}$$

$$0 \leq y_t \leq 1 \text{ for all } t \tag{12}$$

In the following section, we use the framework developed by [13] to construct a randomized algorithm for the problem at hand. The algorithm is based on the primal-dual formulation developed in this section.

IV. Fractional algorithm

A. Algorithm description

The off-line algorithm requires the prior knowledge of $\{A_t^{(2)}\}$. This implies that the algorithm knows at time 0 the values of η_t for all $t \geq 0$. In an on-line setting, at each time $t = 0, 1, \cdots$, the algorithm only knows the value of $\eta_{t'}$ for $t' \leq t$. Thus, the on-line algorithm does not know in advance all the constraints specified by Equation (8), they are revealed to the algorithm one constraint per time slot.

The purpose of **Algorithm 1** is to compute a feasible fractional solution to the primary problem in an on-line fashion. At each slot t, the algorithm obtains the value of $A_t^{(2)}$ which indicates whether an adversary decided to send a new packet to q_2. The algorithm computes the value of z_t, as well as updates the current value of x such that constraint that corresponds to slot t in the primal program is satisfied. The output of the algorithm is used by **Algorithm 2** (described below) which computes a randomized integral policy at each time interval.

Since the design of both algorithms follow the primal-dual framework presented in [13] we present only a sketch of the algorithms.

Algorithm 1 uses additional variables $\tilde{x}$, $\{x_i\}_{i=1}^{\mathcal{K}}$, and $\tilde{z}_t$ for each time t. The constant a in Line 13 is chosen to make the dual solution feasible. For each new time slot t and the associated η_t, only $\{x_i\}_{i=\eta_t+1}^{\mathcal{K}}$ are updated while $\{x_i\}_{i=1}^{\eta_t}$ remain the same. The extra vector of the variables $\{\chi_t, t = 0, 1, \cdots\}$ records the value of x at slot t, and it is used later on by **Algorithm 2**.

B. Correctness proof

Lemma 1. ***Algorithm 1*** *produces a feasible fractional solution to primal program.*

Proof: Let t be an arbitrary time slot, we show that the constraint specified by Equation (8) that corresponds to t in the primary program (i.e., $x + z_t \geq \mathcal{K} - \eta_t$) is satisfied. We consider two cases:

- Case 1: $\tilde{x} \geq 1$. Then,

$$x = \sum_{i=1}^{\mathcal{K}} x_i \geq (\mathcal{K} - \eta_t)\tilde{x} \geq \mathcal{K} - \eta_t. \tag{13}$$

- Case 2: $\tilde{x} < 1$. Then, let $\bar{x}$ be the value of $\tilde{x}$ at time $t-1$, and we get

$$\begin{aligned} x + z_t &= \sum_{i=1}^{\mathcal{K}} x_i + (\mathcal{K} - \eta_t)(1 - \bar{x}) && (14)\\ &\geq (\mathcal{K} - \eta_t)\tilde{x} + (\mathcal{K} - \eta_t)(1 - \bar{x}) && (15)\\ &\geq \mathcal{K} - \eta_t. && (16) \end{aligned}$$

Here, the first inequality is due to the "for" loop in Line 14. The second inequality is due to the fact that $\tilde{x}$ can only increase as the algorithm proceeds. ∎

Lemma 2. ***Algorithm 1*** *produces a feasible fractional solution to the dual program when* $a = (1 + \frac{1}{C_T})^{C_T} - 1$.

Proof: To satisfy the dual constraint (11), the iterative updating process starting from Line 6 should not be allowed to run more than C_T times since y_t is assigned to be 1 at time t. Thus, it suffices to show that $\tilde{x} \geq 1$ after at most C_T time slots. Note that the increment of $\tilde{x}$ forms a geometric sequence with the ratio $1 + 1/C_T$, and hence at the $(C_T - 1)^{\text{th}}$ time slot

---

Algorithm 1 Fractional Primal-Dual algorithm

```
1: z_t, y_t, η_t ← 0 for all t
2: x, x̃ ← 0
3: x_1, ··· , x_K ← 0
4: z̃_t ← 0 for all t
5: a ← (1 + 1/C_T)^{C_T} − 1
   For each new time slot t, the parameters are updated as following.
6: if x̃ < 1 then
7:    if A_t^(2) = 1 then
8:       η_t ← η_{t−1} + 1
9:    else
10:      η_t ← η_{t−1}
11:   end if
12:   z̃_t ← 1 − x̃
13:   x̃ ← x̃(1 + 1/C_T) + 1/(aC_T)
14:   for i = η_t + 1 to K do
15:      x_i ← x̃
16:   end for
17:   x ← Σ_{i=1}^{K} x_i
18:   z_t ← (K − η_t) z̃_t
19:   y_t ← 1
20:   χ_t ← x
21: else
22:   Terminate
23: end if
```

it holds that

$$\tilde{x} = \frac{(1+\frac{1}{C_T})^{C_T} - 1}{a}. \quad (17)$$

Therefore, the dual solutions are feasible when

$$a = (1+\frac{1}{C_T})^{C_T} - 1.$$

■

Theorem 3. *Let OPT be the optimal off-line solution to primal program. Then, the cost of the solution computed by **Algorithm 1** is upper bounded by*

$$(1+\frac{1}{(1+1/C_T)^{C_T}-1})OPT.$$

Proof: By P and D, we denote the values of objective functions of the primal and dual problem respectively. Let ΔP and ΔD be the change of the primal and dual cost between the successive updates. Let Δx be the change of x between two successive time slot. Given time t with the associated η_t, we obtain

$$\begin{aligned}\Delta P &= C_T \Delta x + z_t &(18)\\ &= C_T(\mathcal{K}-\eta_t)(\frac{\tilde{x}}{C_T} + \frac{1}{aC_T}) + (\mathcal{K}-\eta_t)(1-\tilde{x}) &(19)\\ &= (\mathcal{K}-\eta_t)(1+1/a). &(20)\end{aligned}$$

It is easy to verify that $\Delta D = (\mathcal{K}-\eta_t)y_t = \mathcal{K}-\eta_t$. Therefore, $P = (1+1/a)D$ and then $P \le (1+1/a)OPT$ by the weak duality property, where

$$a = (1+\frac{1}{C_T})^{C_T} - 1.$$

■

V. Randomized algorithm

A. Algorithm Description

We use the fractional solution produced by **Algorithm 1** to construct a randomized algorithm, **Algorithm 2** for the original problem. The expected cost of the randomized algorithm is less than or equal to the primal cost of **Algorithm 1**. In particular, **Algorithm 2** uses the values of χ_t (output of **Algorithm 1**) to derive the probability to transmit uncoded packets. At any time t, at least $\lfloor \chi_t \rfloor$ packets need to be transmitted without coding. The algorithm uses variable τ to keep the record of the number of the uncoded packets transmitted. Clearly, if there is one packet from the adversary (i.e., $A_t^{(2)} = 1$ in Line 3), the relay transmits a coded packet.

In Line 6, if $\lfloor \chi_t \rfloor - \tau \ge 1$, then $\lfloor \chi_t \rfloor - \tau$ packets are transmitted immediately. Lines 9 and 11 compute probability for the relay to transmit a packet in q_1 in two different cases. Once the relay decides to transmit a packet without coding, then τ is increased by one in Line 17.

B. Example

To show the operation of Algorithms 1 and 2 we give an example of their execution. Assume the transmission cost

Algorithm 2 Randomized algorithm

```
Require: the value of χ_t from Algorithm 1
 1: p_t ← 0 for all t
 2: τ ← 0

    For each new time slot t, if there is a packet in q_1, do:
 3: if A_t^(2) = 1 then
 4:    transmit one coded packet by combining packets from q_1 and q_2
 5: end if
 6: if ⌊χ_t⌋ − τ ≥ 1 then
 7:    transmit ⌊χ_t⌋ − τ packets from q_1
 8:    τ ← ⌊χ_t⌋
 9:    p_t ← χ_t − τ
10: else if χ_t > τ then
11:    p_t ← (χ_t − χ_{t−1})/[1 − (χ_t − τ)]
12: end if
13: if p_t > 0 then
14:    Choose α uniformly random in from [0, 1]
15:    if α ≤ p_t then
16:       transmit 1 packet in q_1
17:       τ ← τ + 1
18:    end if
19: end if
```

$C_T = 5$ and the initial number of packets in q_1 is $\mathcal{K} = 3$. Table I shows the values computed by Algorithms 1 and 2. The arrivals to q_2 are indicated by $A_t^{(2)}$. For each time slot, the values of $\tilde{x}$, x_1, x_2, x_3 and χ_t are computed by **Algorithm 1**. **Algorithm 2** uses the value of χ_t to calculate the probability p_t of transmission. At time 0, $p_0 = 0.39$ means the relay transmits a packet with the probability 0.39, while "no" in Decision row shows that the final decision is to wait at time 0. At time 1, $\chi_0 = 0.9$ implies that the cumulative probability to transmit a packet without coding is 0.9, i.e., the relay decides to transmit at time 2 with the probability $(0.9-0.39)/(1-0.39) = 0.84$. At that time slot, the relay decides to transmit, therefore τ is updated to 1. At time 2, $\chi_2 = 1.47$ suggests that one packet is supposed to have been transmitted while another packet can be transmitted with the probability $p_2 = 0.47$. At time 3, a coded packet would be transmitted due to an arrival from the adversary. Moreover, only x_2 and x_3 are updated and another packet in q_1 would be transmitted with the probability 0.87.

C. Correctness proof

Theorem 4. ***Algorithm 2** achieves a competitive ratio of*

$$1+\frac{1}{(1+1/C_T)^{C_T}-1}.$$

The algorithm is asymptotically $e/(e-1)$-competitive as $C_T \to \infty$.

Proof: It suffices to show that the expected primal cost of the solution computed by **Algorithm 2** is smaller than the

Time t	0	1	2	3
$A_t^{(2)}$	0	0	0	1
$\tilde{x}$	0.13	0.3	0.49	0.72
x_1	0.13	0.3	0.49	0.49
x_2	0.13	0.3	0.49	0.72
x_3	0.13	0.3	0.49	0.72
χ_t	0.39	0.9	1.47	1.93
p_t	0.39	0.84	0.47	0.87
Decision	no	yes	no	yes
τ	0	1	1	2

TABLE I
A EXAMPLE OF EXECUTION OF ALGORITHMS 1 AND 2.

fractional solution computed by **Algorithm 1**. First notice that at slot t the expected number of transmissions without coding is equal to the number χ_t computed by **Algorithm 1**.

Given the time slot t and the associated η_t, the expected number of packets waiting in q_1 is $\mathcal{K} - \eta_t - \chi_t$, which is less than z_t obtained from **Algorithm 1** since

$$\mathcal{K} - \eta_t - \chi_t = \mathcal{K} - \eta_t - \sum_{i=1}^{\mathcal{K}} x_i \tag{21}$$

$$\leq \mathcal{K} - \eta_t - (\mathcal{K} - \eta_t)\tilde{x} \tag{22}$$

$$\leq \mathcal{K} - \eta_t - (\mathcal{K} - \eta_t)\bar{x} \tag{23}$$

$$= (\mathcal{K} - \eta_t)\tilde{z}_t, \tag{24}$$

where $\bar{x}$ is equal to the value of $\tilde{x}$ at the previous iteration.

We conclude that the expected primal cost from **Algorithm 2** is less than that from **Algorithm 1**. ■

The next lemma shows that in **Algorithm 2** the relay transmits at most four packets in its mini-slot.

Lemma 5. *In **Algorithm 2**, the maximum numbers of packets transmitted in one time slot is 4.*

Proof: It is sufficient to show that $\Delta x \leq 3$. Since the successive increments of $\tilde{x}$ are a geometric sequence, we have

$$\Delta x \leq \frac{\mathcal{K}}{a \cdot C_T}(1 + \frac{1}{C_T})^{C_T - 1}. \tag{25}$$

Notice that $(1 + 1/C_T)^{C_T-1} \leq 3$ and $a \geq 1$. Then, $\Delta x \leq 3$ due to the assumption that $\mathcal{K} \leq C_T$. ■

VI. CONCLUSION

In this paper we investigate the basic trade-off between the total number of transmissions and the waiting time in one-hop wireless coding networks.

We present an on-line algorithm that minimizes the sum of the transmission cost and waiting cost in the worst-case scenario (with respect to the arrivals). The algorithm is based on the Primal-Dual techniques, as well as the randomization approach and archives the expected competitive ratio of $e/(e-1)$. We believe that our results can be extended to a general reverse carpooling setting with multiple relay nodes.

REFERENCES

[1] M. Effros, T. Ho, and S. Kim, "A tiling approach to network code design for wireless networks," in *Information Theory Workshop, 2006. ITW '06 Punta del Este. IEEE*, march 2006, pp. 62 –66.

[2] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[3] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, 2000.

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 497–510, 2008.

[5] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," *Proc. of ACM SIGCOMM*, pp. 169–180, 2007.

[6] Y. E. Sagduyu and A. Ephremides, "Cross-Layer Optimization of MAC and Network Coding in Wireless Queueing Tandem Networks," *IEEE Trans. Inf. Theory*, vol. 54, pp. 554–571, 2008.

[7] A. Khreishah, C. C. Wang, and N. Shroff, "Cross-Layer Optimization for Wireless Multihop Networks with Pairwise Intersession Network Coding," *IEEE J. Sel. Areas Commun.*, vol. 27, pp. 606–621, 2009.

[8] V. Reddy, S. Shakkottai, A. Sprintson, and N. Gautam, "Multipath Wireless Network Coding: A Population Game Perspective," *Proc of IEEE INFOCOM*, 2010.

[9] A. Eryilmaz, D. S. Lun, and B. T. Swapna, "Control of Multi-Hop Communication Networks for Inter-Session Network Coding," *IEEE Trans. Inf. Theory*, vol. 57, pp. 1092–1110, 2011.

[10] T. Ho and H. Viswanathan, "Dynamic Algorithms for Multicast With Intra-Session Network Coding," *IEEE Trans. Inf. Theory*, vol. 55, pp. 797–815, 2009.

[11] Y. Xi and E. M. Yeh, "Distributed Algorithms for Minimum Cost Multicast With Network Coding," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 379–392, 2010.

[12] Y. P. Hsu, N. Abedini, S. Ramasamy, N. Gautam, A. Sprintson, and S. Shakkottai, "Opportunities for Network Coding: To Wait or Not to Wait," *Proc. of IEEE ISIT*, pp. 791–795, 2011.

[13] N. Buchbinder and J. Naor, *The Design of Competitive Online Algorithms via a Primal–Dual Approach*. Hanover, Mass, 2009.