Delay-Aware Wireless Network Coding in Adversarial Traffic

Yu-Pin Hsu, Member, IEEE

Abstract—We analyze a wireless line network employing wireless network coding. The two end nodes exchange their packets through relays. While a packet at a relay might not find its coding pair upon arrival, a transmission cost can be reduced by waiting for coding with a packet from the other side. To strike a balance between the reduced transmission cost and the cost incurred by the delay, a scheduling algorithm determining either to transmit an uncoded packet or to wait for coding is needed. Because of highly uncertain traffic injections, scheduling with no assumption of the traffic is critical. This paper proposes a randomized online scheduling algorithm for a relay in arbitrary traffic, which can be non-stationary or adversarial. The expected total cost (including a transmission cost and a delay cost) incurred by the proposed algorithm is at most $\frac{c}{c_\min} \approx 1.58$ times the minimum achievable total cost. In particular, the proposed algorithm is universal in the sense that the ratio is independent of the traffic. With the universality, the proposed algorithm can be implemented at each relay distributedly (in a multi-relay network) with the same ratio. Moreover, the proposed algorithm turns out to generalize the classic ski-rental online algorithm.

Index Terms—Network coding, scheduling algorithms, competitive analysis.

I. INTRODUCTION

There has been a dramatic proliferation of research on wireless network coding. The wireless network coding can substantially reduce the number of transmissions by exploiting the broadcast nature of wireless medium, resulting in power saving. Illustrated in Fig. 1-(a), two end nodes $n_1$ and $n_2$ exchange their respective packets $p_1$ and $p_2$ belonging to the Galois field $GF(2)$ through a relay. The conventional communication technique (without network coding) requires four transmissions (two for each packet). Leveraging the wireless network coding, only three transmissions are required; precisely, nodes $n_1$ and $n_2$ send packets $p_1$ and $p_2$ to the relay, and the relay broadcasts the scalar-linear combination $p_1 + p_2$ (by bitwise XOR over $GF(2)$) to both end nodes. Each end node $n_i$ (for $i = 1, 2$) can recover its desired packet $p_{i-1}$ by subtracting (over $GF(2)$) packet $p_i$ it already has from packet $p_1 + p_2$ it receives. In general, the wireless network coding can save up to 50% of transmissions as long as the number of relays in a line network in Fig. 1-(b) (as also called the reverse carpooling [2]) is large.

To benefit from the wireless network coding, a relay has to create sufficient coded packets; however, a coded packet at the relay can be created only when packets from both sides are available. Precisely, a relay in Fig. 1-(b) maintains two queues $Q_1$ and $Q_2$ storing packets from both sides, respectively. If both queues are non-empty, then the relay can construct coded packets by combining packets from both queues. However, what should the relay do if only one queue is non-empty? Should the relay wait for coding in the future or just transmit uncoded packets from the non-empty queue? To fully realize the advantage of the wireless network coding would incur packet delays, whereas always transmitting uncoded packets to minimize the delays causes a larger number of transmissions. Therefore, a scheduling algorithm for determining when to code is crucial.

The scheduling problem for a single-relay network as in Fig. 1-(a) under stationary stochastic traffic has been investigated leveraging stochastic control techniques, like Lyapunov theory (e.g., [3]) or Markov decision processes (e.g., [4]). All the prior solutions fail to generalize to non-stationary or adversarial (worst-case) traffic. In particular, they cannot be implemented at each relay distributedly in a multi-relay network with provable performance guarantees. However, non-stationary or adversarial traffic has gained increasing importance in recent years. On one hand, external...
traffic injections at nodes $n_1$ or $n_2$ can arbitrarily be generated by their sources, following no particular probabilistic assumption. On the other hand, the relay cannot expect the scheduling algorithms employed by nodes $n_1$ and $n_2$ to follow a stationary probabilistic distribution. In particular, [5] claimed that the adversarial traffic is a better traffic model. Because of those practical issues, the research on the adversarial traffic has attracted much attention in recent years (e.g., [6]). Although network coding design for adversarial channels has been an active area (e.g., [7]), little attention was given to the adversarial traffic in network-coding-enabled networks. To fill the gap, this paper aims to develop a universal scheduling algorithm for arbitrary traffic with a provable performance guarantee.

Moreover, note that the ski-rental problem [8] is a classic problem in an adversarial setting, where for each day a skier decides either to buy a ski or to continue renting a ski without knowing the skier’s last vacation, e.g., the day when the snow melts. The ski-rental setting has been exploited in several works (e.g., [9]) for managing delays under some uncertainties. This paper shows that the proposed scheduling algorithm can solve generalized ski-rental scenarios.

A. Contributions

Our main contribution lies in designing and analyzing scheduling for delay-aware wireless network coding in the adversarial traffic. The objective is to minimize a total cost, including a transmission cost and a delay cost, for each relay. To reach the goal, we show that our problem can be cast into a linear program. Leveraging primal-dual techniques [10] for the linear program, we propose a randomized online scheduling algorithm for each relay in a multi-relay network. In particular, the proposed algorithm can guarantee that the worst-case ratio between the expected total cost incurred by the proposed online algorithm and that incurred by an optimal offline algorithm is (asymptotically) $\frac{1}{1.58}$. In addition to the theoretical worst-case analysis, the average-case analysis is conducted via computer simulations. Moreover, it turns out that the proposed algorithm can generalize the classic ski-rental algorithm to several scenarios.

B. Related Works

Scheduling design for network-coding-enabled networks has been extensively explored from various perspectives. Most scheduling works with network coding aimed to maximize throughput (i.e., stability regions), e.g., [11]–[20]. While [11]–[14] considered static network environments and solved deterministic optimization problems, [15]–[20] considered dynamic network environments and solved dynamic optimal control problems. Moreover, several scheduling works with network coding investigated delays, e.g., [21]–[23]. In addition to the throughput or delays, some prior works analyzed other utilities or constraints when network coding is enabled, e.g., [24] maximized a video reception quality and [25] considered a Quality-of-Service (QoS) constraint.

The most relevant works on the trade-off between delays and power consumption (with wireless network coding) in relay networks are [3], [4]. While [3] proposed a scheduling algorithm using Lyapunov techniques, [4] showed the optimality of a threshold-type scheduling algorithm using Markov decision processes. All those prior solutions were based on stochastic models with some stationary assumptions but cannot apply to non-stationary settings, especially in multi-relay networks. In contrast, we explore the trade-off in non-stationary settings.

II. System Overview

A. Network Model

Consider a wireless line network in Fig. 1-(b). The two end nodes $n_1$ and $n_2$ send $N_1$ and $N_2$ packets in $GF(2)$, respectively, to each other through shared relay nodes. Divide time into slots and index them by $t = 1, 2, \cdots$. Suppose that a perfect schedule of wireless links is given, so that during each slot each node can transmit some packets under a transmission constraint without any interference. The interference-free link schedule can be achieved by existing medium access control (MAC) protocols, e.g., scheduled TDMA used in [4], CSMA proposed by [26], or coded ALOHA proposed by [27]. In fact, our design can work with any MAC protocol (see Remark 14 later).

Consider a relay in Fig. 1-(b). The relay constructs queues for storing those packets that arrive at the relay but cannot be transmitted immediately upon arrival. As shown in Fig. 1-(b), the relay maintains two queues $Q_1$ and $Q_2$ for packets generated by nodes $n_1$ and $n_2$, respectively. At the beginning of each slot $t$, there are $A_1(t)$ new packets arriving at queue $Q_1$ and $A_2(t)$ new packets arriving at queue $Q_2$. By $A = \{(A_1(1), A_2(1)), (A_1(2), A_2(2)), \cdots\}$ we define an arrival pattern for the relay. The arrival pattern depends on the number of packets transmitted in the previous slot by its neighboring nodes. The arrival pattern is arbitrary, which can be non-stationary or even adversarial.

Let $Q_1(t)$ and $Q_2(t)$ be the number of packets at queues $Q_1$ and $Q_2$, respectively, immediately after the packet arrivals in slot $t$. If $Q_1(t) = 0$ and $Q_2(t) = 0$, then the relay idles in slot $t$. If $Q_1(t) \neq 0$ and $Q_2(t) \neq 0$, then the relay transmits some coded packets (under the transmission constraint) by combining (over $GF(2)$) packets from both queues. Transmitting the coded packets can save the number of transmissions (without incurring any delay).

1To save the number of transmissions, both neighboring nodes of the relay must be able to decode the coded packets transmitted by the relay. To that end, we leverage the reverse carpooling technique [2]. Each node (including both end nodes and all relay nodes) keeps packets it transmitted previously for a while, so that when it receives a coded packet, it can decode the coded packet. See Footnote 2 later for the amount of time to keep a packet it previously transmitted. Moreover, each relay employs the decode-and-forward mechanism, where it decodes before re-encoding and transmitting packets.
minimizes the number of transmissions by coding. To strike a balance between the delays and the number of transmissions, the best decision is unclear when exactly one of the queues is non-empty.

To investigate the best decision for each slot, we let $D(t)$ be the relay’s decision on the number of packets (including both uncoded and coded packets) transmitted in slot $t$. We assume that the broadcast channel from the relay to its neighboring nodes is noiseless. This simple model facilitates to explore the delays for coding in the arbitrary arrival pattern. In fact, our design can extend to adversarial ON-OFF channels (see Remark 24 later). Under the noiseless assumption, the queuing dynamics is

$$Q_i(t + 1) = \max\{Q_i(t) - D(t), 0\} + A_i(t + 1),$$

for all $i$ and $t$. For example, if $Q_1(1) = 5$, $Q_2(1) = 3$, and $D(1) = 4$, then the relay transmits three coded packets combining three packets from queues $Q_1$ and $Q_2$ each, and transmits one uncoded packet from queue $Q_1$; moreover, if $A_1(2) = 2$, then $Q_1(2) = 3$.

A scheduling algorithm $\pi = \{D(1), D(2), \cdots\}$ for the relay specifies decision $D(t)$ for each slot $t$. A scheduling algorithm is called an offline scheduling algorithm if arrival pattern $A$ is given as a prior. In contrast, a scheduling algorithm is called an online scheduling algorithm if arrival pattern $A$ (along with the numbers $N_1$ and $N_2$ of packets) is unavailable; instead, it knows the present arrivals $A_1(t)$ and $A_2(t)$ only, for each slot $t$.

**B. Problem Formulation**

To capture the trade-off between the delays and the number of transmissions, we define a holding cost and a transmission cost as follows. Suppose that holding a packet at the end of a slot incurs a cost of one unit. Moreover, suppose that each packet transmission takes a constant cost of $C$ units, where we assume that transmitting a coded packet incurs the same transmission cost as transmitting an uncoded packet. See Remark 6 for non-consistent costs for transmitting coded and uncoded packets. Moreover, we consider the case when the value of $C$ is greater than one.

Given arrival pattern $A$, we define a total cost $J(A, \pi)$ under scheduling algorithm $\pi$ by

$$J(A, \pi) = \sum_{t=1}^{\infty} C \cdot D(t) + \max\{Q_1(t) - D(t), 0\} + \max\{Q_2(t) - D(t), 0\},$$

where the first term $C \cdot D(t)$ reflects the cost of transmitting $D(t)$ packets in slot $t$ and the other terms $\max\{Q_1(t) - D(t), 0\} + \max\{Q_2(t) - D(t), 0\}$ reflects the cost of delaying all remaining packets for one slot. Since we consider the finite numbers $N_1$ and $N_2$, the minimum achievable total cost is finite.

We aim to develop an online scheduling algorithm such that the total cost is minimized for all possible arrival patterns $A$. However, without knowing arrival pattern $A$ (along with the total numbers $N_1$ and $N_2$ of packets) in advance, an online scheduling algorithm is unlikely to achieve the minimum total cost (obtained by an optimal offline scheduling algorithm). We characterize our online scheduling algorithm in terms of the competitiveness against an optimal offline scheduling algorithm, defined as follows.

**Definition 1:** For arrival pattern $A$, let $OPT(A) = \min_{\pi} J(A, \pi)$ be the minimum total cost for all possible (offline) scheduling algorithms $\pi$. Then, an online scheduling algorithm $\pi$ is called $\gamma$-competitive if

$$J(A, \pi) \leq \gamma \cdot OPT(A),$$

for all possible arrival patterns $A$, where $\gamma$ is called the competitive ratio of the online scheduling algorithm $\pi$.

**Remark 2:** A $\gamma$-competitive online scheduling algorithm guarantees that the resulting total cost is at most $\gamma$ times the minimum total cost, regardless of arrival patterns $A$. Thus, while a $\gamma$-competitive online scheduling algorithm can be implemented at each relay in the multi-relay network in a distributed way, it guarantees the competitive ratio $\gamma$ for each relay.

We aim to design and analyze an online scheduling algorithm for minimizing the competitive ratio.

**III. ONE-SIDED ADVERSARIAL TRAFFIC**

We start with a fixed number of packets waiting for coding; in particular, this section focuses on the following setting:

1. Queue $Q_1$ has all $N_1$ (with $N_1 \leq C$) packets initially, i.e., $A_1(t) = N_1$ for all $t \geq 2$.
2. Queue $Q_2$ is injected by arbitrary traffic with a total of $N_2$ packets.
3. The relay can transmit any number of packets in each slot.

The setting is referred to as the one-sided adversarial traffic. With the first and second assumptions, we can focus on a fixed number $N_1$ of packets at queue $Q_1$ waiting for coding, while capturing the key feature of the adversarial arrival pattern at queue $Q_2$. In fact, the first assumption is practical as well for bursty traffic at queue $Q_1$. Section IV will also generalize to two-sided adversarial traffic. Note that, under the first assumption, the relay never delays the packets at queue $Q_2$ for minimizing the total cost. The third assumption is made for delivering a clear insight into our innovation. Lemma 15 will analyze the maximum number of transmissions required by the proposed online scheduling algorithm; moreover, Section IV-E will extend to a transmission constraint.
A. Overview of Our Methodology

This section provides an overview of our methodology (leveraging primal-dual techniques [10] for linear programs):

1) We propose linear program (4) for optimally solving our scheduling problem in the offline fashion (with arrival pattern \(A\) as a prior).
2) We propose Alg. 1 for sub-optimally solving linear program (4) in the online fashion (without arrival pattern \(A\) as a prior).
3) We analyze the objective value of (linear program (4)) computed by Alg. 1 through a solution (produced also by Alg. 1) to the dual of the linear program. We show that the objective value computed by Alg. 1 is no more than \(\frac{e}{e-1}\) times that dual objective value. Then, the duality theory yields that the objective value computed by Alg. 1 is no more than \(\frac{e}{e-1}\) times the minimum objective value (of linear program (4)).
4) By transforming the fractional solution produced by Alg. 1 to randomized decisions, we propose a randomized online scheduling algorithm in Alg. 2.
5) We show that the expected cost incurred by Alg. 2 is no more than the objective value computed by Alg. 1. Then, by the third bullet, the expected total cost incurred by Alg. 2 is also no more than \(\frac{e}{e-1}\) times the minimum objective value (i.e., minimum achievable total cost).

Section III-B formulates the linear program (a primal program) and its dual program. While Section III-C proposes Alg. 1 for solving the primal program and the dual program in the online fashion, Section III-D analyzes the solution produced by Alg. 1. Leveraging the solution produced by Alg. 1, Section III-E proposes Alg. 2 for solving our scheduling problem and analyzes its expected total cost.

B. Primal-Dual Formulation

Given arrival pattern \(A\), this section casts the offline scheduling problem (under the one-sided adversarial traffic) into a linear program. To that end, we introduce some variables:

- \(x\): the number of packets at queue \(Q_1\) transmitted without coding.
- \(z(t)\): the number of packets at queue \(Q_1\) at the end of slot \(t\).

If the relay decides to transmit \(x\) uncoded packets at queue \(Q_1\), then it must\(^*\) transmit the \(x\) uncoded packets in slot 1. Thus, the total cost in Eq. (1) under the one-sided adversarial traffic can be expressed by

\[
J(A, \pi) = C \cdot N_2 + C \cdot x + \sum_{t=1}^{\infty} z(t), \tag{2}
\]

where the first term \(C \cdot N_2\) is the cost of transmitting all packets at queue \(Q_2\), the second term \(C \cdot x\) is the cost of transmitting the \(x\) uncoded packets at queue \(Q_1\) in slot 1 (i.e., transmitting coded packets at queue \(Q_1\) is free), and the last term \(\sum_{t=1}^{\infty} z(t)\) is the cost incurred by holding the \(N_1 - x\) packets at queue \(Q_1\).

By removing the constant \(C \cdot N_2\) from Eq. (2), we have the following scheduling problem.

Problem 3: Under the one-sided adversarial traffic, develop a scheduling algorithm for the packets at queue \(Q_1\) such that the cost \(C \cdot x + \sum_{t=1}^{\infty} z(t)\) is minimized.

Remark 4: This remark shows that the classic ski-rental problem [8] is a special case of our Problem 3. In the ski-rental problem, a skier arrives at a resort on day 1 with no ski. For each day, the skier decides either to buy a ski or to rent a ski. If the skier buys a ski in a day, then the skier does not have to rent a ski after that day. While renting a ski for a day takes one dollar, buying a ski takes \(C\) dollars. The skier will stay at the resort for \(T\) days until the last vacation day. The goal is to minimize the buying cost plus the total renting cost. Given the instance of the ski-rental problem, we construct an instance of our Problem 3. We construct one packet for each queue, i.e., \(N_1 = N_2 = 1\). We construct a packet staying at queue \(Q_1\) in slot 1 (corresponding to the skier). We construct a packet arriving at queue \(Q_2\) in slot \(T\) (corresponding to the last vacation day). Next, we link a skier’s decision with a relay’s decision. While the skier rents a ski on day \(t\) if and only if the relay idles in slot \(t\), the skier buys a ski on day \(t\) if and only if the relay transmits the packet at queue \(Q_1\) without coding in slot \(t\). While the skier does not have to make decisions after day \(T\), the relay also does not have to make decisions after slot \(T\) (because the relay can transmit a coded packet in slot \(T\) if the packet at \(Q_1\) still stays at that queue in slot \(T\)). With the link between the ski-rental problem and our Problem 3, variable \(x\) in Problem 3 can indicate if the skier buys a ski, and variable \(z(t)\) in Problem 3 can indicate if the skier rents a ski on day \(t\). Suppose that holding the packet at queue \(Q_1\) for a slot takes one dollar, and that transmitting an uncoded packet from queue \(Q_1\) takes \(C\) dollars. Then, the value of \(C \cdot x + \sum_{t=1}^{\infty} z(t)\) in Problem 3 can represent the buying cost plus the total renting cost. Thus, the ski-rental problem equivalently becomes our Problem 3. In other words, the ski-rental problem is a special case (\(N_1 = N_2 = 1\)) of our Problem 3.

Remark 5: Following Remark 4, this remark shows that our Problem 3 is a generalization of the ski-rental problem. We can think of each packet at queue \(Q_1\) as a skier and think of a slot when a packet arrives at queue \(Q_2\) as the day when a skier has to leave. Moreover, buying a ski takes \(C\) dollars while renting a ski for a day takes one dollar. With the transformation, Problem 3 considers a group of skiers (i.e., the \(N_1\) packets at queue \(Q_1\)) with potentially different last vacation days (i.e., the arriving slots at queue \(Q_2\)). Those skiers cooperatively make a buying or renting decision on each day for minimizing the total buying cost plus the total renting cost.

Remark 6: If transmitting a coded packet incurs a cost of \(C_1\) units and transmitting an uncoded packet incurs a different cost of \(C_2\) units with \(C_1 > C_2\), then the total cost in
Eq. (1) becomes
\[ J(A, \pi) = C_2 \cdot N_2 + C_2 \cdot x + (C_1 - C_2)(N_1 - x) + \sum_{t=1}^{\infty} z(t), \]
where the term \((C_1 - C_2)(N_1 - x)\) is the extra cost for transmitting the coded packets. Then, we can replace cost \(C\) in Problem 3 with \(2C_2 - C_1\). Note that \(2C_2 - C_1 \geq 0\). If \(C_1\) were higher than \(2C_2\), then transmitting a coded packet by combining two packets would not save any cost from transmitting two uncoded packets. The rest of the paper focuses on the constant cost \(C\) without loss of generality.

Next, we propose the following integer program for optimally solving Problem 3 in the offline fashion:

**Integer Program:**

\[
\begin{align*}
\text{min } & \quad C \cdot x + \sum_{t=1}^{\infty} z(t) \\
\text{s.t. } & \quad x + z(t) \geq N_1 - n_2(t) \text{ for all } t; \\
& \quad x, z(t) \in \mathbb{N} \text{ for all } t,
\end{align*}
\]

where \(n_2(t) = \sum_{\tau=t}^{\infty} A_2(\tau)\) is the total number of packets arriving at queue \(Q_2\) until slot \(t\). The constraint in Eq. (3b) is because for each slot \(t\) the number of packets at queue \(Q_1\) is at least \(N_1 - x - n_2(t)\), where \(x\) packets at queue \(Q_1\) are transmitted without coding in slot \(t = 1\) and at most \(n_2(t)\) packets at queue \(Q_1\) are transmitted with coding by slot \(t\).

Next, by relaxing the integrality constraint in Eq. (3c) to real numbers, we obtain the following linear program.

**Linear Program (Primal Program):**

\[
\begin{align*}
\text{min } & \quad C \cdot x + \sum_{t=1}^{\infty} z(t) \\
\text{s.t. } & \quad x + z(t) \geq N_1 - n_2(t) \text{ for all } t; \\
& \quad x, z(t) \geq 0 \text{ for all } t.
\end{align*}
\]

After the relaxation, a feasible fractional solution for \(x\) in linear program (4) can no longer represent a decision for the number of packets at queue \(Q_1\) transmitted without coding (but an integral solution for \(x\) in linear program (4) can). In fact, the next lemma shows that the relaxation has no integrality gap.

**Lemma 7:** The relaxation from integer program (3) to linear program (4) has no integrality gap.

**Proof:** Suppose that an optimal solution to linear program (4) is non-integral. Then, we establish a contradiction. See our technical report [28] for details.

From Lemma 7, Problem 3 can be optimally solved in polynomial time if arrival pattern \(A\) is given in advance: Solve for variable \(x\) in linear program (4) then, transmit \(x\) uncoded packets in slot 1. After transmitting the uncoded packets in slot 1, all other packets at queue \(Q_1\) always wait for packets at queue \(Q_2\) for coding.

Next, while Section III-B proposes an online algorithm for sub-optimally solving for variable \(x\) without knowing arrival pattern \(A\) in advance, Section III-D analyzes the objective value in Eq. (4a) computed by the proposed online algorithm by its dual program. Thus, we refer to linear program (4) as a primal program and express its dual program as follows.

**Dual Program:**

\[
\begin{align*}
\max & \quad \sum_{t=1}^{\infty} (N_1 - n_2(t))w(t) \\
\text{s.t. } & \quad \sum_{t=1}^{\infty} w(t) \leq C; \\
& \quad 0 \leq w(t) \leq 1 \text{ for all } t.
\end{align*}
\]

**C. Primal-Dual Algorithm**

This section proposes a primal-dual algorithm in Alg. 1 for obtaining a solution to primal program (4) and dual program (5). The primal-dual algorithm does not have arrival pattern \(A\) as a prior; instead, it can obtain the present arrivals \(A_1(t)\) and \(A_2(t)\) only, for each slot \(t\).

**Algorithm 1:** Primal-Dual Algorithm for Solving Primal Program (4) and Dual Program (5)

```plaintext
/* Initialize all variables at the beginning of slot 1 as follows: */
1 x, z(t), w(t) ← 0 for all t;
2 x_1, ..., x_{N_1}, z_1(t), ..., z_{N_1}(t) ← 0 for all t;
// Auxiliary variables.
3 θ ← (1 + 1/|C|) - 1; // θ is a constant with the function of cost C

/* For each new slot t = 1, 2, ..., the variables are updated as follows: */
4 for i = n_2(t) + 1 to N_1 do
5 if x_i < 1 then
6 \quad z_i(t) ← 1 - x_i;
7 \quad x_i ← x_i(1 + \frac{1}{\theta C}) + \frac{1}{\theta C};
8 \quad w(t) ← 1;
9 end
10 end
12 z(t) ← \sum_{i=1}^{N_1} z_i(t);
13 x ← \sum_{i=1}^{N_1} x_i;
```

Alg. 1 initializes all variables (in Lines 1 and 2) at the beginning of slot 1. Obtaining the present arrivals \(A_1(t)\) and \(A_2(t)\) at the beginning of each new slot \(t\), Alg. 1 updates all variables for slot \(t\). For updating the value of \(x\), Alg. 1 introduces a set of auxiliary variables \(x_1, ..., x_{N_1}\) (initialized in Line 2). The intuition behind updating variable \(x_i\) and \(x\) in Lines 6, 8, and 13 is following: We can imagine the value of \(x_i\) to be a probability of transmitting the \(i\)-th (counted from the head of queue \(Q_1\)) packet at queue \(Q_1\) without coding. Precisely, for each slot \(t\), Line 8 increases the value of \(x_i\) for those packets potentially staying at queue \(Q_1\):

\( \)
A total of \( n_2(t) \) packets arrive at queue \( Q_2 \) by slot \( t \), yielding at most \( n_2(t) \) coded packets until \( t \). As such, Line 5 considers \( x_i \), for \( i = n_2(t) + 1 \) until \( N_1 \), because only the \((n_2(t)+1)\)-th packet until the \( N_1 \)-th packet might wait at queue \( Q_1 \) in slot \( t \), but other packets have been transmitted with coding by slot \( t \).

Moreover, if the value of \( x_i \) is greater than or equal to one (i.e., the condition in Line 6 fails), then the \( i \)-th packet has been transmitted without coding by slot \( t \). Thus, Line 8 updates only those \( x_i \)'s satisfying the condition in Line 6.

The constant \( \theta \) used in Line 8 is specified as the function of transmission cost \( C \) in Line 4 for satisfying the dual constraint in Eq. (5b). Then, Line 13 sets the value of \( \bar{Q} \) subject to the constraints in Eq. (5c).

For proving the lemma, we define the increment (under Alg. 1) of packets at queue \( Q_1 \) by exploiting the solution produced by Alg. 1, Section III-E.

We want to emphasize that the solution produced by Alg. 1 in each slot can be transformed to a probability of \( z_{\lfloor \frac{t}{C} \rfloor} \) for satisfying the constraint in Line 6.

This section analyzes the primal objective value in Eq. (4a) computed by Alg. 1.

Theorem 11: Let \( OPT(4)(A) \) be the minimum objective value in linear program (4). Then, the primal objective value in Eq. (4a) computed by Alg. 1 is bounded above by

\[
\frac{1}{1 + \frac{1}{\theta}(1 + \frac{1}{\theta})|C|} OPT(4)(A),
\]

for all possible arrival patterns \( A \).

Proof: Let \( \Delta \mathcal{D}(t) \) be the increment (under Alg. 1) of the primal objective value in Eq. (4a) in slot \( t \) and let \( \Delta \mathcal{D}(t) \) be that of the dual objective value in Eq. (5a) in slot \( t \).

Appendix B establishes that

\[
\Delta \mathcal{D}(t) \leq \left( 1 + \frac{1}{\theta} \right) \Delta \mathcal{D}(t),
\]

for all \( t \). Let \( \mathcal{D} \) and \( \mathcal{D} \) be the primal and dual objective values, respectively, computed by Alg. 1. Then, \( \mathcal{D} = \sum_{i=1}^{\infty} \Delta \mathcal{D}(t) \) and \( \mathcal{D} = \sum_{i=1}^{\infty} \Delta \mathcal{D}(t) \); therefore, the result follows since

\[
\mathcal{P} \leq \left( 1 + \frac{1}{\theta} \right) \mathcal{D} \leq \left( 1 + \frac{1}{\theta} \right) OPT(4)(A),
\]

where the last inequality is due to the weak duality [10].

E. Randomized Online Scheduling Algorithm

Leveraging Alg. 1, this section proposes a randomized online scheduling algorithm in Alg. 2.

For each slot \( t \), Alg. 2 transmits \( \min(Q_1(t), A_2(t)) \) coded packets (in Line 5) by combing packets left at queue \( Q_1 \) and the new arriving packets at queue \( Q_2 \). Then, to decide whether to transmit uncoded packets for each slot, Lines 6 - 10 and 12 update the values of \( x_i \) and \( x \) in the same way as Alg. 1 does. In addition, Alg. 2 uses another variable \( x_{\text{pre}} \) (in Line 11) to record the value of \( x \) at the beginning (before update in Line 12) of each slot. Let \( \bar{x}_{\text{pre}}(t) \) be the value of \( x_{\text{pre}} \) at the end of slot \( t \).

At the beginning of slot 1, Line 4 chooses a random number \( u \in [0, 1) \) from a continuous uniform distribution between 0 and 1. According to Lines 13 - 20, if there exists a \( k \in \mathbb{N} \)
such that \( u + k \in [\bar{x}_{\text{pre}}(t), \bar{x}(t)] \), then the relay transmits an uncoded packet in slot \( t \). Note that, if there are multiple \( k \)'s such that \( u + k \in [\bar{x}_{\text{pre}}(t), \bar{x}(t)] \), then the relay transmits multiple uncoded packets in slot \( t \), until the present value of \( u \) is greater than or equal to \( \bar{x}(t) \) (as in Line 18).

Let \( \Delta \bar{x}(t) = \bar{x}(t) - \bar{x}(t) = \bar{x}(t) - \bar{x}_{\text{pre}}(t) \) be the increment of the value of \( x \) in slot \( t \). The idea behind Alg. 2 is that, with the random choice of \( u \), the expected number of uncoded packets transmitted in slot \( t \) is exactly \( \Delta \bar{x}(t) \).

**Theorem 12:** The expected competitive ratio of Alg. 2 is

\[
1 + \frac{1}{(1 + \frac{1}{C})(C) - 1},
\]

approaching \( \frac{e}{C} \) as \( C \) tends to infinity.

**Proof:** We show that the expected cost of transmitting uncoded packets by Alg. 2 is \( C \cdot \sum_{t=1}^{\infty} \Delta \bar{x}(t) = C \cdot \bar{x}(\infty) \), which is the value of the first term in Eq. (4a) computed by Alg. 1. Moreover, we show that the expected number of packets left at queue \( Q_1 \) at the end of slot \( t \) under Alg. 2 is less than or equal to \( \bar{z}(t) \), which is the value of the second term in Eq. (4a) computed by Alg. 1. Thus, the expected cost incurred by Alg. 2 is less than or equal to the primal objective value in Eq. (4a) computed by Alg. 1. Then, the result immediately follows from Theorem 11. See Appendix C for details.

Remark 13: Recall that a competitive ratio is the worst-case ratio for all possible cases (i.e., arrival patterns \( A \)) and recall that the ski-rental problem is a case of our Problem 3 (from Remark 4). Thus, the minimum achievable competitive ratio for our problem is no higher than that for the ski-rental problem. Because the minimum achievable competitive ratio for the ski-rental problem is \( \frac{1}{2} \) [8] and Alg. 2 can also achieve that competitive ratio, we can conclude that Alg. 2 achieves the minimum achievable competitive for Problem 3.

Remark 14: We want to emphasize that the competitive ratio in Theorem 12 is independent of arrival patterns \( A \), i.e., regardless of the MAC protocol. Thus, Alg. 2 can be implemented at each relay in the multiple-relay network; meanwhile, it can ensure the same competitiveness for each relay.

The next lemma investigates the maximum number of uncoded packets per slot required by Alg. 2.

**Lemma 15:** Alg. 2 transmits at most three uncoded packets in each slot.

**Proof:** Since \( \{\Delta \bar{x}_1(1), \Delta \bar{x}_1(2), \cdots, \Delta \bar{x}_1(C)\} \) is the geometric sequence with the initial value of \( \frac{1}{\theta C} \) and the ratio of \( 1 + \frac{1}{\theta} \), we have

\[
\Delta \bar{x}(t) = \sum_{i=1}^{N_1} \Delta \bar{x}_i(t) \leq N_1 \frac{\theta C}{1 + \frac{1}{\theta}}(1 - C^{-1}).
\]

Moreover, because of \( (1 + \frac{1}{\theta})^{C-1} \leq 3 \), \( \theta \geq 1 \), and \( N_1 \leq C \) (from the assumption for the one-sided traffic), we have \( \Delta \bar{x}(t) \leq 3 \). Thus, at most three \( k \)'s such that \( u + k \in [\bar{x}_{\text{pre}}(t), \bar{x}(t)] \), i.e., Alg. 2 transmits at most three uncoded packets in each slot.

To analyze the computational complexity of Alg. 2, we note that there are at most \( N_1 \) iterations in Lines 6 - 10. Moreover, there are at most 3 iterations in Lines 13 - 20 (by Lemma 15). Since \( N_1 \leq C \) (from the assumption for the one-sided traffic), the computational complexity of Alg. 2 is \( O(C) \). As the value of \( C \) grows, the computational complexity increases but the competitive ratio in Theorem 12 decreases.

### IV. Two-Sided Adversarial Traffic

This section relaxes the first assumption in the one-sided adversarial traffic by allowing arbitrary traffic at both queues \( Q_1 \) and \( Q_2 \). We start with the scenario where only packets at a queue can wait for coding; in particular, this section starts with the following setting:

1. The packets at queue \( Q_1 \) can wait for coding but those at queue \( Q_2 \) are transmitted immediately upon arrival.
2. The relay can transmit any number of packets in each slot.

This setting is referred to as the two-sided adversarial traffic. This model can make us focus on decisions for a queue while capturing the key feature of the two-sided adversarial traffic. In fact, the first assumption is practical as well when the traffic generated by node \( n_2 \) is urgent and even cannot delay for more than one slot (e.g., urgent events in intelligent transportation systems or ultra-reliable low-latency communications (URLLC) [29] in 5G). Later, Section IV-D will relax the first assumption by extending to the general case...
when packets at both queues can wait for coding. Moreover, Section VI-E will relax the second assumption by imposing a transmission constraint.

We introduce some variables similar to Section III:

- $x_i$: indicate if the $i$-th packet at queue $Q_1$ is transmitted without coding upon arrival, where $x_i = 1$ if the packet is transmitted without coding; $x_i = 0$ otherwise;
- $z(t)$: the number of packets at queue $Q_1$ at the end of slot $t$.

We have the following problem similar to Problem 3.

Problem 16: Under the two-sided adversarial traffic, develop a scheduling algorithm for the packets at queue $Q_1$ such that the cost $C \cdot \sum_{i=1}^{N} x_i + \sum_{t=1}^{\infty} z(t)$ is minimized.

Remark 17: Following the argument in Remark 5, Problem 16 considers a group of skiers arriving arbitrarily with potentially different last vacation days. Those skiers cooperatively make a buying or renting decision in each day for minimizing the total buying cost plus the total renting cost.

Section VI-A discusses ideas underlying another primal-dual formulation that will be proposed by Section VI-B for solving Problem 16. With the new primal-dual formulation, Section VI-C proposes a primal-dual algorithm for solving Problem 16 in the online fashion.

A. Ideas Underlying the Primal-Dual Formulation

The next example shows that an immediate extension from linear program (4) along with Alg. 1 cannot solve Problem 16 with the competitive ratio in Theorem 11.

Example 18: Suppose that two packets arrive at queue $Q_1$ in slots 1 and 3, respectively, and no packet arrives at queue $Q_2$. Assume transmission cost $C = 2$. In this case, the optimal solution to Problem 16 is $x_1 = 1$ and $x_2 = 1$, i.e., both packets at queue $Q_1$ are optimally transmitted without coding upon arrival. In particular, the optimal solution satisfies the following linear program (similar to linear program (4)).

Linear Program (Primal Program):

\[
\begin{align*}
\text{min} \quad & 2(x_1 + x_2) + \sum_{t=1}^{\infty} z(t) \\
\text{s.t.} \quad & x_1 + z(t) \geq 1 \text{ for } t = 1, 2; \\
& x_1 + x_2 + z(t) \geq 2 \text{ for } t = 3, 4, \cdots; \\
& x_1, x_2, z(t) \geq 0 \text{ for all } t.
\end{align*}
\]

The associated dual program can be expressed as

Dual Program:

\[
\begin{align*}
\text{max} \quad & \sum_{t=1}^{2} w(t) + 2 \cdot \sum_{t=3}^{\infty} w(t) \\
\text{s.t.} \quad & \sum_{t=1}^{\infty} w(t) \leq 2; \\
& 0 \leq w(t) \leq 1 \text{ for all } t.
\end{align*}
\]

Applying the idea behind Alg. 1, we would update $x_i \leftarrow x_i(1 + \frac{1}{\theta}) + \frac{1}{\theta} w(t)$ and update $w(t) \leftarrow 1$ until the dual constraint in Eq. (7b) becomes tight. Given $C = 2$, the constant $\theta$ is $(1 + \frac{1}{\theta})^2 - 1 = \frac{\theta}{2}$. In slot 1, update $x_1$ to be $\frac{1}{1 + \frac{1}{\theta}} = \frac{\theta}{2}$ and update $w(1)$ to be one. In slot 2, update $x_2$ to be $\frac{\theta}{2}(1 + \frac{1}{\theta}) + \frac{1}{\theta} = 1$ and update $w(2)$ to be one. Because the dual constraint in Eq. (7b) becomes tight in slot 2, we cannot update any variable since slot 3; in particular, we cannot update $x_3$ when the second packet arrives at queue $Q_1$. Thus, the second packet waits forever, yielding an infinite holding cost.

To tackle the issue in the above example, the next example proposes another primal-dual formulation.

Example 19: Note that an optimal solution for $x_1$ and $x_2$ in linear program (6) also satisfies the following linear program, where we use $z_1(t)$ and $z_2(t)$ to indicate if the first packet and second packet, respectively, stay at queue $Q_1$ at the end of slot $t$.

Linear Program (Primal Program):

\[
\begin{align*}
\text{min} \quad & 2(x_1 + x_2) + \sum_{t=1}^{\infty} z_1(t) + \sum_{t=1}^{\infty} z_2(t) \\
\text{s.t.} \quad & x_1 + z_1(t) \geq 1 \text{ for } t = 1, 2, \cdots; \\
& x_2 + z_2(t) \geq 1 \text{ for } t = 3, 4, \cdots; \\
& x_1, x_2, z_1(t), z_2(t) \geq 0 \text{ for all } t.
\end{align*}
\]

While expressing variable $z(t)$ in Eq. (6a) by $z_1(t) + z_2(t)$ in Eq. (8a), we substitute the original constraints in Eqs. (6b) and (6c) by the constraints in Eqs. (8b) and (8c). The associated dual program can be expressed as

Dual Program:

\[
\begin{align*}
\text{min} \quad & \sum_{t=1}^{\infty} w_1(t) + \sum_{t=3}^{\infty} w_2(t) \\
\text{s.t.} \quad & \sum_{t=1}^{\infty} w_1(t) \leq 2; \\
& \sum_{t=3}^{\infty} w_2(t) \leq 2; \\
& 0 \leq w_1(t), w_2(t) \leq 1 \text{ for all } t.
\end{align*}
\]

Follow the idea behind Alg. 1 as discussed in Example 18. In slot 1, update $x_1$ to be $\frac{\theta}{2}$ and update $w_1(1)$ to be one. In slot 2, update $x_1$ to be one and update $w_1(2)$ to be one. In slot 3, update $x_2$ to be $\frac{\theta}{2}$ and update $w_2(3)$ to be one. In slot 4, update $x_2$ to be one and update $w_2(4)$ to be one. The updating process can achieve the competitive ratio in Theorem 11.

The above example implies that the idea of Alg. 1 can solve Problem 16 with the same competitive ratio, if we can formulate a linear program with constraints for each individual packet (like Eqs. (8b) and (8c)) instead of those for all arriving packets (like Eqs. (6b) and (6c)). In this context, we introduce additional variables: let $z_i(t)$ indicate if the $i$-th packet stays at queue $Q_1$ at the end slot $t$, where $z_i(t) = 1$ if it does and $z_i(t) = 0$ otherwise. For each slot $t$, the value of $x_i + z_i(t)$ is either zero or one, where $x_i + z_i(t) = 0$ implies that the $i$-th packet at queue $Q_1$ is transmitted with coding by slot $t$ and $x_i + z_i(t) = 1$ implies that the packet is either transmitted without coding by slot $t$ or stays at queue $Q_1$ at the end of slot $t$. By the next example, we emphasize that the constraints should be carefully considered.
Example 20: Suppose that two packets arrive at queue $Q_1$ in slots 1 and 2, respectively, and one packet arrives at queue $Q_2$ in slot 3. Assume transmission cost $C = 4$. In this case, the optimal solution to Problem 16 is $x_1 = 1$ and $x_2 = 0$. Next, given the optimal decision for the packet at queue $Q_2$ (i.e., optimally transmitted with coding), we consider constraints for each packet at queue $Q_1$, as follows:

- **Slot $t = 1$:** A packet arrives at queue $Q_1$ in slot 1. Thus, we can obtain $x_1 + z_1(1) = 1$.
- **Slot $t = 2$:** The other packet arrives at queue $Q_1$ in slot 2. Thus, we can obtain $x_1 + z_1(2) = 1$ and $x_2 + z_2(1) = 1$.
- **Slot $t = 3$:** A packet arrives at queue $Q_2$. Since we are given that the packet at queue $Q_2$ optimally codes with a packet at queue $Q_1$, two options are following: (1) $x_1 + z_1(3) = 0, x_2 + z_2(3) = 1$, i.e., the first packet at queue $Q_1$ is transmitted with coding, and the second packet either is transmitted without coding or waits in slot 3; (2) $x_1 + z_1(3) = 1, x_2 + z_2(3) = 0$.

Slot $t > 3$: No packet arrives at both queues. Thus, if $x_1 + z_1(3) = 0$ and $x_2 + z_2(3) = 1$, then $x_1 + z_1(t) = 0$ and $x_2 + z_2(t) = 1$; otherwise, $x_1 + z_1(t) = 1$ and $x_2 + z_2(t) = 0$.

We calculate the minimum value of $4(x_1 + x_2) + \sum_{t=1}^{\infty} z_1(t) + \sum_{t=1}^{\infty} z_2(t)$ subject to the two possible constraints, i.e., forming two different linear programs:

- **Consider the constraints of** $x_1 + z_1(t) \geq 1$ **for** $1 \leq t \leq 2$, **and** $x_2 + z_2(t) \geq 1$ **for** $t \geq 2$: The optimal solution is $x_1 = 0$ and $x_2 = 1$, while the minimum objective value is 6.

- **Consider the constraints of** $x_1 + z_1(t) \geq 1$ **for** $t \geq 1$, **and** $x_2 + z_2(2) \geq 1$: The optimal solution is $x_1 = 1$ and $x_2 = 0$, while the minimum objective value is 5.

Thus, only the second set of constraints is correct. The idea underlying the correct set of constraints is that the first packet waits for a longer time (for coding) than the second packet does.

Let $I(t) = \{i : x_i + z_i(t) \geq 1\}$ be the set of indices such that the value of $x_i + z_i(t)$ in slot $t$ is specified to be greater than or equal to one. Let $I = \{I(1), I(2), \cdots\}$. Our goal is to identify a correct set $I$ of constraints such that the solution to minimize the cost (in Problem 16) subject to the set $I$ is an optimal solution to Problem 16. Example 20 suggests that, when a packet arrives at queue $Q_2$ in slot $t$, a correct set $I(t)$ of constraints in slot $t$ can be obtained by removing the most recent packet in set $I(t-1)$ of the previous slot. The argument will be confirmed in the next section.

B. Primal-Dual Formulation

With the idea developed in Example 20, we propose an algorithm in Alg. 3 for identifying a correct set $I$ of constraints. Line 2 initiates set $I(t)$ in slot $t$ to be $I(t-1)$ of the previous slot. When a packet arrives at queue $Q_1$ in slot $t$, Line 4 adds the corresponding index to set $I(t)$. Line 6 introduces a variable $q_2$ to indicate the available packets at queue $Q_2$ for coding; precisely, Line 6 sets the value of variable $q_2$ to be the present arrivals $A_2(t)$ at queue $Q_2$. Since Line 7, if $q_2 \neq 0$ (i.e., there is a packet at queue $Q_2$) and $I(t) \neq 0$ (i.e., there is a packet at queue $Q_1$), then Line 9 removes index $i^*$ (i.e., the most recent packet in set $I(t)$ as in Line 8) from set $I(t)$ and Line 10 removes one packet from queue $Q_2$.

We formulate a linear program subject to the set $I$ produced by Alg. 3 as follows.

**Linear Program (Primal Program):**

\[
\begin{align*}
\min & \quad C \cdot \sum_{i=1}^{N_1} x_i + \sum_{t=1}^{\infty} \sum_{i=1}^{N_1} z_i(t) \\
\text{s.t.} & \quad x_i + z_i(t) \geq 1 \quad \text{for all } i \in I(t) \text{ and } t. \quad (10a) \\
& \quad 0 \leq w_i(t) \leq 1 \quad \text{for all } i \text{ and } t. \quad (11c)
\end{align*}
\]

The next theorem establishes that linear program (10) can optimally solve Problem 16.

**Theorem 21:** The solution to linear program (10) is an optimal solution to Problem 16.

**Proof:** We prove by induction. See our technical report [28].

The dual program of primal program (10) is following.

**Dual Program:**

\[
\begin{align*}
\max & \quad \sum_{t=1}^{\infty} \sum_{i \in I(t)} w_i(t) \\
\text{s.t.} & \quad \sum_{t \in I(t)} w_i(t) \leq C \quad \text{for all } i; \quad (11b) \\
& \quad 0 \leq w_i(t) \leq 1 \quad \text{for all } i \text{ and } t. \quad (11c)
\end{align*}
\]

C. Primal-Dual Algorithm

Note that Alg. 3 can learn a correct set $I(t)$ of constraints for each slot $t$ in the online fashion. Leveraging the online feature, we develop a primal-dual algorithm in Alg. 4 for solving Problem 16 in the online fashion. For each slot $t$, Alg. 4 updates those $x_i$’s in the set $I(t)$ in Lines 14 - 20. The updating process is similar to that in Alg. 1.

Using the same arguments as those in the proofs of Lemmas 8 and 9, the next lemma establishes the feasibility of the solution produced by Alg. 4.

**Lemma 22:** Alg. 4 produces a feasible solution to primal program (10) and dual program (11).
Algorithm 4: Primal-Dual Algorithm for Solving Primal Program (10) and Dual Program (11)

```java
/* Initialize all variables at the beginning of slot 1 as follows: */
1 i, z_i(t), w_i(t) ← 0 for all i and t;
2 θ ← (1 + 1/C) - 1;
3 I(t) ← ∅ for all t;
/* For each new slot t = 1, 2, · · · , the variables are updated as follows: */
4 I(t) ← I(t - 1);
5 forall the i-th packet arriving at queue Q_1 in slot t do
6 I(t) ← I(t) ∪ {i};
7 end
8 q_2 ← A_2(t);
9 while q_2 ≠ 0 and I(t) ≠ ∅ do
10 i* ← max I(t);
11 I(t) ← I(t) - {i*};
12 q_2 ← q_2 - 1;
13 end
14 forall the i ∈ I(t) do
15 if x_i < 1 then
16 z_i(t) ← 1 - x_i;
17 x_i ← x_i(1 + C/(θC)) + 1/(θC);
18 w_i(t) ← 1;
19 end
20 end
```

Similar to Theorem 11, the next theorem shows that Alg. 4 can achieve the same competitive ratio as Alg. 1 does.

**Theorem 23:** Let OPT_{10}(A) be the minimum objective value in linear program (10). Then, the primal objective value in Eq. (10a) computed by Alg. 4 is bounded above by

\[
(1 + \frac{1}{(1 + \frac{1}{C})^C - 1})OPT_{10}(A),
\]

for all possible arrival patterns A.

**Proof:** See Appendix D.

Then, similar to Alg. 2, we can transform the solution produced by Alg. 4 to a randomized online scheduling algorithm for managing the delay-award coding decision at queue Q_1. In particular, the scheduling algorithm can also achieve the same expected competitive ratio as that in Theorem 12, approaching \( \varepsilon^- \) when cost C is large enough.

### D. Scheduling Both Queues

This section extends Alg. 4 to the case when both queues Q_1 and Q_2 can wait for each other. In this context, we propose a waiting-coding queueing system consisting of a waiting queue Q_w and a coding queue Q_c at the relay. While queue Q_w stores those packets that can wait for coding, queue Q_c stores those packets that can find coding pairs at the waiting queue immediately upon arrival.

Precisely, let Q_w(t) and Q_c(t) be the number of packets at queue Q_w and queue Q_c, respectively, at the end of slot t. If the Q_w(t - 1) packets at queue Q_w belong to queue Q_1, then the A_1(t) (i.e., the number of packets arriving at the original queue Q_1) new arriving packets enter queue Q_w at the beginning of slot t, and

1) if Q_w(t - 1) + A_1(t) ≥ A_2(t), then the A_2(t) new arriving packets enter queue Q_c at the beginning of slot t;
2) if Q_w(t - 1) + A_1(t) < A_2(t), then only Q_w(t - 1) + A_1(t) out of the A_2(t) new arriving packets enter queue Q_c at the beginning of slot t, but the remaining A_2(t) - (Q_w(t) + A_1(t)) packets enter queue Q_w at the beginning of slot t.

In contrast, if the Q_w(t - 1) packets at queue Q_w belong to queue Q_2, then the waiting-coding queueing system operates in the opposite way. In other words, while packets entering queue Q_c are transmitted (with coding) immediately upon arrival, packets entering queue Q_c need scheduling decisions.

With the transformation, the waiting-coding queueing system becomes the previously discussed model where only packets at queue Q_w can wait for coding. Thus, the randomized online scheduling algorithm associated with Alg. 4 can apply to the waiting-coding queueing system with the expected competitive ratio in Theorem 12. Furthermore, Section V will demonstrate the superiorities of the proposed scheduling algorithm and the proposed waiting-coding queueing system via computer simulations.

### E. A Transmission Constraint

Recall that Alg. 4 might transmit more than one packet in a slot (but less than three uncoded packets, as shown in Lemma 15). This Section considers a transmit constraint: the relay can transmit at most one packet in each slot. According to Section IV-D, we can focus on scheduling packets at queue Q_1 while all packets at queue Q_2 are transmitted immediately upon arrival.

Under the transmission constraint, if more than one packet arrive at a queue, then those additional packets (except for one of them) cannot be processed in the arriving slot for any scheduling algorithm. Thus, without loss of generality, we can further assume that at most one packet can arrive at each queue in each slot. If more than one packet arrives at a queue, we can just move them to the following slots, so that at most one packet arrives at that queue. With that assumption, we analyze the number of uncoded packets required by the randomized online scheduling algorithm (like Alg. 2) associated with Alg. 4: following the proof of Lemma 15, the number of uncoded packets transmitted in slot t is

\[
\sum_{i ∈ I(t)} p_{i}, \Delta x_i(t) ≤ \sum_{j=1}^{[C]} \frac{1}{θC(1 + \frac{1}{C})^j} = 1,
\]

where the inequality is because: (1) at most \([C]\) packets (as in the proof of Lemma 9) in set I(t) that can be updated by Line 17 of Alg. 4 in slot t; (2) the j-th most recent packet in set I(t) has been updated by Line 17 of Alg. 4 for at least j times since its arrival; (3) the value of \(\Delta x_i(t)\) is \(\frac{1}{θC(1 + \frac{1}{C})^j}\) if the i-th packet is updated by Line 17 of Alg. 4 for j times.

We emphasize that, by the above analysis, the randomized online scheduling algorithm might need two transmissions in
a slot, i.e., one potential coded packet plus one potential uncoded packet. To make the randomized online scheduling algorithm perform under the constraint of at most one transmission, we revise Alg. 4 as follows: the updates in Lines 14 - 20 perform only when no packet arrives at queue $Q_2$. That is because, if a packet arrives at queue $Q_2$, the relay has to transmit a coded packet; thus, stop updating those variables for transmitting an uncoded packet. Following the line in [30, Theorem 5], the randomized online scheduling algorithm associated with the revised Alg. 4 can also achieve the same expected competitive ratio of $1$ when cost $C$ is large enough. Moreover, Section V will validate the revised randomized online scheduling algorithm via computer simulations.

Remark 24: We remark that the revised randomized online scheduling algorithm can also solve the adversarial ON-OFF channel, also by stopping updating when the channel is OFF.

V. NUMERICAL STUDIES

We have analyzed the proposed randomized online scheduling algorithm in the worst-case scenario; in contrast, we investigate the proposed algorithm in the average-case scenario by computer simulations in this section.

First, we simulate a single-relay network (as in Fig. 1-(a)) where packets arrive at queues $Q_1$ and $Q_2$ according to the i.i.d. Bernoulli distributions with means $\max\{\min\{P_1, 1\}, 0\}$ and $\max\{\min\{P_2, 1\}, 0\}$, respectively, where $P_1$ and $P_2$ are the Gaussian random variables (for adding some noises to the Bernoulli arrivals) with means $p_1$ and $p_2$, respectively, and variance $\sigma^2$. Moreover, the relay can transmit at most one packet for each slot. We compare the proposed scheduling algorithm (i.e., the randomized online scheduling algorithm associated with Alg. 4 along with the waiting-coding queueing system in Section IV-D and the stopping mechanism in Section IV-E) with threshold-type scheduling algorithms, where the relay transmits an uncoded packet in a slot if (in the original queueing system) a queue is empty and the non-empty queue size is over its threshold in that slot. The optimized-threshold scheduling algorithm was proposed in [4] for minimizing the long-run average cost in the stochastic environment. However, deriving an optimal threshold for each queue needs the statistics $p_1$ and $p_2$, i.e., the optimized-threshold scheduling algorithm is an offline scheduling algorithm. Fig. 2 displays the ratio between the total cost (in 10,000 slots) incurred by the proposed scheduling algorithm and that incurred by the optimized-threshold scheduling algorithm. We can observe that the ratio for the proposed scheduling algorithm is at most 1.35 (in Fig. 2-(a) when $p_2 = 0.1$ and $C = 10$). That is, the proposed algorithm performs much better than what we analyzed in the worst-case scenario (with the expected competitive ratio of $\frac{C}{C-1} \approx 1.58$). In addition, Fig. 2 also displays the ratio between the total cost incurred by the $C$-threshold scheduling algorithm and that incurred by the optimized-threshold scheduling algorithm, where the $C$-threshold is an online scheduling algorithm with the constant threshold $C$ and was analyzed in [3]. According to Fig. 2, our algorithm significantly outperforms the $C$-threshold scheduling algorithm. Moreover, We can observe that the ratio (for a fixed $p_2$ and a fixed $C$) decreases as the variance increases. That is because the ratio in Fig. 2-(a) decreases when the expected arrival rate $p_2$ at queue $Q_2$ moves toward 0.5 and the expected arrival rate at queue $Q_2$ in Figs. 2-(b) and 2-(c) (i.e., $E[\max\{\min\{P_2, 1\}, 0\}]$) moves toward 0.5 (because of the truncation of the Gaussian variable $P_2$ to 0 and 1) when the variance increases.

Second, we investigate the coding overheads incurred by the three scheduling algorithms. Fig 3 displays the number of coded packets when $\sigma^2 = 0$ (i.e., for the case in Fig. 2-(a)).
We can observe that while the proposed scheduling algorithm yields less coded packets than the optimized-threshold scheduling algorithm, the $C$-threshold scheduling algorithm yields more than that. That is, while the proposed scheduling algorithm is a little conservative (in waiting for coding), the $C$-threshold scheduling algorithm waits too long. That is why the proposed scheduling algorithm and the $C$-threshold scheduling algorithm cannot minimize the total cost.

Third, we simulate multi-relay networks where external packets arrive at the two end relays according to the i.i.d. Bernoulli distributions. While each relay can transmit at most one packet for each slot, a received packet from the other relay in a slot cannot be processed until the next slot. Fig. 4-(a) displays the ratio of total costs (with respect to the optimized-threshold scheduling algorithm) when there are two relays and both relays take transmission costs $C_1$ and $C_2$, respectively. The optimized-threshold scheduling algorithm identifies a threshold for each queue by exhaustive search for minimizing the total cost among all possible thresholds. We want to emphasize that an optimal scheduling for the two-relay network is still unclear. In particular, the optimized-threshold scheduling algorithm might not minimize the long-run average cost in this case, though its great performance has been demonstrated in [4] by computer simulations. We can observe that the ratios for the proposed scheduling algorithm and the $C$-threshold scheduling algorithm in Fig. 2-(a) and Fig. 4-(a) are almost the same. In addition, Fig. 4-(a) also displays the ratio for the sub-optimized-threshold scheduling algorithm, which identifies a threshold for each queue by exhaustive search for minimizing the total cost subject to the condition that all left queues have the same threshold and all right queues do as well. We can observe that the sub-optimized-threshold scheduling algorithm can achieve almost the same total cost as the optimized-threshold scheduling algorithm does. Thus, we compared the proposed scheduling algorithm with the sub-optimized-threshold scheduling algorithm when there are more than two relays and all relays take the same transmission cost $C$. Figs. 4-(b) and Figs. 4-(c) displays the ratio with respect to the sub-optimized-threshold scheduling algorithm. We can observe that the ratio is insensitive to the numbers of relays; in particular, the proposed scheduling algorithm still significantly outperforms the $C$-threshold scheduling algorithm.

VI. CONCLUDING REMARKS

In this paper, we treated a wireless line network employing wireless network coding. The inherent trade-off between packet delays and transmission power consumption under adversarial traffic was studied. In particular, we developed a randomized online scheduling algorithm. The proposed scheduling algorithm not only can theoretically guarantee the expected competitive ratio of $\frac{e}{2} \approx 1.58$ for each relay, but also can numerically approach the minimum total cost (including a delay cost and a transmission cost) by computer simulations; moreover, the proposed scheduling algorithm can solve more general ski-rental settings.

While this paper focused on line networks, some discussions on extending to more general networks are following. Consider a relay with multiple line networks traversing, where the two end nodes of each line network exchange their packets. If the relay can transmit one packet in each slot for each line network (as in Section IV-E), then the proposed scheduling algorithm can immediately apply to each line network individually. However, if a relay has a transmission constraint on the total number of transmissions for all line networks, then linear program (10) needs another constraint for specifying that the total number of transmissions cannot be over that transmission constraint. That is an interesting future work. To solve the problem, the prior work [31] (considering a “box constraint”) might be helpful.

Some problems are still open as follows. This paper focused on the worst-case analysis. To theoretically analyze the proposed algorithm in the average-case scenario is interesting and can help understand why it has a great performance in the simulation results. Moreover, we analyzed the competitive ratio of the proposed algorithm in the waiting-coding queuing system; however, the competitive ratio in the original queuing system is still undiscovered. Finally, a MAC protocol is given to this paper. Joint scheduling design of MAC and coding would be a promising future topic.

APPENDIX A
PROOF OF LEMMA 8

First, the primal constraint in Eq. (4c) holds obviously because Alg. 1 initializes all variables to be zeros in Lines 1 and 2 and never decreases their values. Second, the primal constraint in Eq. (4b) holds for each slot $t$ as follows:

![Fig. 4. (a) Ratio versus $p_2$ (fixed $p_1 = 0.5$, $\sigma^2 = 0$ and $C_1 = 5$) when there are two relays; (b) Ratio with respect to the sub-optimized-threshold scheduling algorithm versus the number of relays (fixed $p_1 = 0.5$, $p_2 = 0.1$, $\sigma^2 = 0$); (c) Ratio with respect to the sub-optimized-threshold scheduling algorithm versus the number of relays (fixed $p_1 = 0.5$, $p_2 = 0.9$, $\sigma^2 = 0$).](image-url)
First, we compare the expected cost of transmitting uncoded packets by Alg. 2 with the term $C \cdot \bar{x}(\infty)$ of the primal objective value in Eq. (4a) computed by Alg. 1. Note that, for a given $u$, there must exist $\lfloor \Delta \bar{x}(t) \rfloor k$’s such that $u + k \in [\bar{x}_{\text{pre}}(t), \bar{x}(t))$, i.e., Alg. 2 transmits $\lceil \Delta \bar{x}(t) \rceil$ uncoded packets in slot $t$; in addition, according to [30], Alg. 2 transmits one more uncoded packet with probability $\Delta \bar{x}(t) - \lfloor \Delta \bar{x}(t) \rfloor$. Thus, the expected number of uncoded packets transmitted by Alg. 2 in slot $t$ is $\Delta \bar{x}(t)$; moreover, the expected total number of uncoded packets transmitted by Alg. 2 is $\sum_{t=1}^{\infty} \Delta \bar{x}(t) = \bar{x}(\infty)$. We can obtain that the expected cost of transmitting uncoded packets by Alg. 2 is $C \cdot \bar{x}(\infty)$, which is exactly the value of the first term of the primal objective value in Eq. (4a) computed by Alg. 1.

Second, we compare the expected number of packets left at queue $Q_1$ at the end of slot $t$ under Alg 2 with the term $\bar{z}(t)$ of the primal objective value in Eq. (4a) computed by Alg. 1. Note that the expected number of packets left at queue $Q_1$ at the end of slot $t$ under Alg 2 is

$$\max \{ N_1 - \sum_{\tau=1}^{t} \Delta \bar{x}(\tau), 0 \}$$

$$= \max \{ N_1 - n_2(t) - \bar{x}(t), 0 \}.$$ 

1) If $\bar{x}_i(t) < 1$ for $i = n_2(t) + 1, \cdots, N_1$, then the expected number of packets left at queue $Q_1$ at the end of slot $t$ under Alg 2 is less than the term $\bar{z}(t)$ of the primal objective value in Eq. (4a) computed by Alg 1 because $N_1 - n_2(t) - \bar{x}(t)$ is less than or equal to $\bar{z}(t)$ (by the primal feasibility in Eq. (4b) of Alg. 1).

2) If $\bar{x}_i(t) \geq 1$ for $i = n_2(t) + 1, \cdots, N_1$, then both the expected number of packets left at queue $Q_1$ at the end of slot $t$ under Alg 2 and the term $\bar{z}(t)$ of the primal objective value in Eq. (4a) computed by Alg 1 are zeros because $N_1 - n_2(t) - \bar{x}(t) \leq N_1 - n_2(t) - \sum_{i=n_2(t)+1}^{N_1} \bar{x}_i(t) \leq 0$.

We conclude that the expected cost in Problem 3 incurred by Alg. 2 is less than or equal to the primal objective value in Eq. (4a) computed by Alg 1. Then, the result immediately follows from Theorem 11.
Eq. (10a) (under Alg. 4) is
\[
\Delta \mathcal{P}(t) = \sum_{i=1}^{N_1} C_i \cdot (\hat{x}_i(t) - \bar{x}_i(t)) + \bar{z}_i(t) = \sum_{i \in \mathcal{I}(t) \setminus \{i : \hat{x}_i(t) \geq 1\}} C_i \cdot \left( \frac{\hat{x}_i(t)}{C} + \frac{1}{\theta} \cdot \frac{1}{C} \right) + (1 - \hat{x}_i(t)) = |\mathcal{I}(t) - \{i : \hat{x}_i(t) \geq 1\}|(1 + \frac{1}{\theta}).
\]
Moreover, the change of the dual objective value in Eq. (11a) (under Alg. 4) is
\[
\Delta \mathcal{P}(t) = |\mathcal{I}(t) - \{i : \hat{x}_i(t) \geq 1\}|\bar{w}(t) = |\mathcal{I}(t) - \{i : \hat{x}_i(t) \geq 1\}|.
\]
Then, following the line in the proof of Theorem 11 yields the result.

REFERENCES

[1] Y.-P. Hsu and A. Sprintson, “Opportunistic network coding: Competitive analysis,” in Proc. IEEE NetCod, Jun. 2012, pp. 191–196.

[2] M. Effros, T. Ho, and S. Kim, “A tiling approach to network code design for wireless networks,” in Proc. IEEE ITW, Mar. 2006, pp. 62–66.

[3] E. N. Cifcioglu, Y. E. Sagduyu, R. A. Berry, and A. Yener, “Cost-delay tradeoffs for two-way relay networks,” IEEE Trans. Wireless Commun., vol. 10, no. 12, pp. 4100–4109, Dec. 2011.

[4] Y.-P. Hsu, N. Abedini, N. Gautam, A. Sprintson, and S. Shakkottai, “Opportunities for network coding: To wait or not to wait,” IEEE/ACM Trans. Netw., vol. 23, no. 6, pp. 1876–1899, Dec. 2015.

[5] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson, “Adversarial queueing theory,” Proc. ACM STOC, vol. 96, 1996, pp. 376–385.

[6] Q. Liang and E. Modiano, “Network utility maximization in adversarial environments,” in Proc. IEEE INFOCOM, Apr. 2018, pp. 594–602.

[7] A. Ravagnani and F. R. Kschischang, “Adversarial network coding,” IEEE Trans. Inf. Theory, vol. 65, no. 1, pp. 198–219, Jan. 2019.

[8] A. R. Karlin, C. Kenyon, and D. Randall, “Dynamic TCP acknowledgement and other stories about $e/(e-1)$,” in Proc. ACM STOC, 2001, pp. 502–509.

[9] G. Lee, W. Saad, M. Bennis, A. Mehbdiniya, and F. Adachi, “Online ski rental for ON/OFF scheduling of energy harvesting base stations,” IEEE Trans. Wireless Commun., vol. 16, no. 5, pp. 2976–2990, May 2017.

[10] N. Buchbinder and J. S. Naor, The Design Competitive Online Algorithms Via a Primal-Dual Approach, vol. 3, nos. 2–3. Boston, MA, USA: Now, 2009.

[11] A. ParandehGheibi, A. Ozdaglar, M. Effros, and M. Médard, “Optimal reverse carpooling over wireless networks—A distributed optimization approach,” in Proc. IEEE CISS, Mar. 2010, pp. 1–6.

[12] D. Traskov, M. Heindlmaier, M. Médard, and R. Koetter, “Scheduling for network-coded multicast,” IEEE/ACM Trans. Netw., vol. 20, no. 5, pp. 1479–1488, Oct. 2012.

[13] T. Wiese, M. Riemenschneider, and W. Utschick, “Scheduling for network-coded multicast with interference,” IEEE/ACM Trans. Netw., vol. 20, no. 5, pp. 1479–1488, Oct. 2012.

[14] T. Wiese, M. Riemenschneider, and W. Utschick, “Scheduling for network-coded multicast with interference,” IEEE/ACM Trans. Netw., vol. 20, no. 5, pp. 1479–1488, Oct. 2012.

[15] H. Seferoglu and A. Markopoulou, “Network coding-aware queue management for TCP flows over coded wireless networks,” IEEE/ACM Trans. Netw., vol. 22, no. 4, pp. 1297–1310, Aug. 2014.

[16] N. Moghaddam, H. Li, H. Zeng, and L. Liu, “Lyapunov scheduling and optimization in network coded wireless multicast systems,” IEEE Trans. Veh. Technol., vol. 67, no. 6, pp. 5135–5145, Jun. 2018.

[17] T. Ho and H. Visswanathan, “Dynamic algorithms for multicast with intra-session network coding,” IEEE Trans. Inf. Theory, vol. 55, no. 2, pp. 797–815, Feb. 2009.

[18] 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, no. 2, pp. 1092–1110, Feb. 2011.

[19] C. Fragiadakis, G. S. Paschos, L. Georgiadis, and L. Tassiulas, “Dynamic wireless network coding with overhearing and variable channel rates,” IEEE J. Sel. Areas Commun., vol. 33, no. 2, pp. 185–198, Feb. 2015.

[20] N. M. Jones, B. Shradar, and E. Modiano, “Distributed CSMA with pairwise coding,” in Proc. IEEE INFOCOM, Apr. 2013, pp. 2094–2102.

[21] I.-H. Hou, “Broadcasting delay-constrained traffic over unreliable wireless links with network coding,” IEEE/ACM Trans. Netw., vol. 23, no. 3, pp. 728–740, Jun. 2015.

[22] M. S. Al-Abiad, A. Douik, and S. Sorour, “Rate aware network codes for cloud radio access networks,” IEEE Trans. Mobile Comput., vol. 18, no. 8, pp. 1898–1910, Aug. 2019.

[23] B. T. Swapna, A. Eryilmaz, and N. B. Shroff, “Throughput-delay analysis of random linear network coding for wireless broadcasting,” IEEE Trans. Inf. Theory, vol. 59, no. 10, pp. 6328–6341, Oct. 2013.

[24] C. Li, H. Xiong, J. Zou, and D. O. Wu, “Joint dynamic rate control and transmission scheduling for scalable video multicast over wireless networks,” IEEE Trans. Multimedia, vol. 20, no. 2, pp. 361–378, Feb. 2018.

[25] M. S. Al-Abiad, M. J. Hossain, and S. Sorour, “Cross-layer cloud offloading with quality of service guarantees in fog-RANs,” IEEE Trans. Commun., vol. 67, no. 12, pp. 8435–8449, Dec. 2019.

[26] J. Ni, B. Tan, and R. Srikant, “Q-CSMA: Queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks,” IEEE/ACM Trans. Netw., vol. 20, no. 3, pp. 825–836, Jun. 2012.

[27] E. Paolini, G. Liva, and M. Chiani, “Coded slotted ALOHA: A graph-based method for uncoordinated multiple access,” IEEE Trans. Inf. Theory, vol. 61, no. 12, pp. 6815–6832, Dec. 2015.

[28] Y.-P. Hsu, “Delay-aware wireless network coding in adversarial traffic,” 2019, arXiv:1911.08078. [Online]. Available: https://arxiv.org/abs/1911.08078

[29] A. Azari, M. Ozger, and C. Cavdar, “Risk-aware resource allocation for URLLC: Challenges and strategies with machine learning,” IEEE Commun. Mag., vol. 57, no. 3, pp. 42–48, Mar. 2019.

[30] Y.-H. Tseng and Y.-P. Hsu, “Online energy-efficient scheduling for timely information downloads in mobile networks,” in Proc. IEEE ISIT, Jul. 2019, pp. 1022–1026.

[31] N. Buchbinder and J. Naor, “Online primal-dual algorithms for covering and packing,” Math. Oper. Res., vol. 34, no. 2, pp. 270–286, May 2009.

Yu-Pin Hsu (Member, IEEE) received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, in 2005 and 2007, respectively, and the Ph.D. degree from Texas A&M University, USA, in 2014. He was a Post-Doctoral Fellow with the Singapore University of Technology and Design, Singapore, in 2014, and with the Massachusetts Institute of Technology, USA, in 2015. In 2016, he joined National Taipei University, Taiwan, where he is currently an Assistant Professor. His research interests include communication networks with focus on algorithmic and control-theoretic aspects.