Coding based Data Broadcasting for Time Critical Applications with Rate Adaptation

Xiumin Wang, Chau Yuen and Yinlong Xu

Abstract—In this paper, we dynamically select the transmission rate and design wireless network coding to improve the quality of services such as delay for time critical applications. In a network coded system, with low transmission rate and hence longer transmission range, more packets may be encoded, which increases the coding opportunity. However, low transmission rate may incur extra transmission delay, which is intolerable for time critical applications. We design a novel joint rate selection and wireless network coding (RSNC) scheme with delay constraint, so as to maximize the total benefit (where we can define the benefit based on the priority or importance of a packet for example) of the packets that are successfully received at the destinations without missing their deadlines. We prove that the proposed problem is NP-hard, and propose a novel graph model to mathematically formulate the problem. For the general case, we prove that there is no polynomial time algorithm to determine the transmission rate and coding strategy for each application. For a special case when all delay constraints are the same, we study the pairwise coding and present a polynomial time pairwise coding algorithm that achieves an approximation ratio of $1 - \frac{1}{e}$ to the optimal pairwise coding solution, where $e$ is the base of the natural logarithm. Finally, simulation results demonstrate the superiority of the proposed RSNC scheme.

I. INTRODUCTION

With the increase in both wireless channel bandwidth and computational capability of wireless devices, wireless networks can be used to support time critical applications such as video streaming or interactive gaming. Such time critical applications require the data content to reach the destination node(s) in a timely fashion, i.e., a delay deadline is imposed on packet reception, beyond which the reception becomes useless (or invalid) [1]–[5]. These constraints can be imposed either by applications or the users. For example, many financial users are interested in the up-to-minute stock quotes so as to react to dynamic and rapid market. As another example, in wireless location-based services, the queried information (e.g., the traffic jam) is valid within a local area, as when the mobile user (e.g., vehicle user) leaves the area, the information becomes useless [6].

Recently, network coding becomes a promising approach to improve wireless network performance [7]–[10]. Specifically, the work in [12] proposed the first network coding based packet forwarding architecture, named COPE, to improve the throughput of wireless networks. With COPE, each node opportunistically overhears some of the packets transmitted by its neighbors, which are not intended to itself. The relay node can then intelligently XOR multiple packets and forward it to multiple next hops with only one transmission, which results in a significant throughput improvement. Another important work on wireless network coding is index coding [17]–[19]. In index coding, a source/server node needs to send some packets over a wireless broadcast channel to some destinations/clients, and initially each destination holds a subset of packets (i.e., side information). Recent works show that with network coding, the number of transmissions required can be reduced significantly, which thus improves the throughput.

In most recent works, network nodes always transmit packets at a fixed rate. However, most wireless systems are now capable of performing adaptive modulation to vary the link transmission rate in response to the signal to interference plus noise at the receivers. Transmission rate diversity exhibits a rate-range tradeoff: the higher the transmission rate, the shorter the transmission range for a given transmission power [20]. To aid overhearing, one may use the lowest transmission rate, so as to successfully deliver packet to more receivers/overhearing nodes. Although this may increase the coding opportunity, it may not yield good performance, especially for time critical applications, as the arrival times of the packets may be delayed due to lower transmission rate.

In the literature, a few works studied the relationships between adapting the transmission rate and the network coding gain [20]–[23]. The work in [20] showed that compared with pure network coding scheme, joint rate adaptation and network coding is more effective in throughput performance. They also proposed a joint rate selection and coding scheme to minimize the sum of the uplink and the downlink costs in star network topology. The work in [21] mathematically formulated the optimal packet coding and rate selection problem as an integer programming problem, and proposed an efficient heuristic algorithm to jointly find a good combination of coding solution and transmission rate. There are only a few works considered the delay guarantee of packet receptions, which is especially important for time critical applications.

So far, the works in [24]–[30] considered the delay constraint of packet reception with network coding. Specifically, [24] designed a jointly scheduling of packet transmission and network coding to meet the restriction of packet receptions in a multi-hop wireless network. Compared with multi-hop transmission, [25] studied a wireless broadcasting scheduling
over a one-hop communication. To meet the hard deadline of the packets, they designed adaptive network coding and formulated the problem as a Markov decision process. However, they assume that all the packets have the same deadline. The work in [26] aimed to optimize the delay of multicasting a data stream from a sender to multiple one-hop receivers with network coding. Based on queuing theory, they analyzed the delay performance from both system and receiver perspectives. The works in [27], [28] also considered the delay constraint of packet receptions over one-hop communication, and proposed a coding scheme to minimize the number of packets that miss their deadlines. All the above literatures worked well in their designed settings. However, they all assume that the transmission rates on all the links are the same and fixed.

Take Fig. 1 as an example, where source node s needs to transmit packet $p_1$, $p_2$, $p_3$ to node $d_1$, $d_2$, $d_3$ respectively. Fig. 1(a) gives the set of overheard packets $H(d_i)$ at destination $d_i$ (i.e. packets overheard by $d_i$ in previous transmissions, and available at $d_i$). Suppose that the size of each packet is $B = 10k$, and the maximum transmission rates from $s$ to $d_1$, $d_2$, $d_3$ are $5k/s$, $2k/s$ and $2k/s$, respectively. Fig. 1(b) shows the reception deadline of each required packet at its destination. For the current transmission, according to the work in [12], [27], $s$ will send the encoded packet $p_1 \oplus p_2 \oplus p_3$, as the most number of destinations can decode it. However, there is a problem for selecting the transmission rate at $s$. If $5k/s$ is selected, $d_2$, $d_3$ cannot successfully receive the packet, as the maximum transmission rates from $s$ to them are both $2k/s$. If $2k/s$ is selected, although all of the three receivers $d_1$, $d_2$, $d_3$ can receive and decode one “wanted” packet, $p_1$ will miss its deadline at $d_1$, as its arrival time is $\frac{10k}{2k} = 5s$. As an alternative, we may choose to first send packet $p_1$ with transmission rate $5k/s$, where destinations $d_3$ will obtain a “wanted” packet in $2s$. After this transmission, the encoded packet $p_2 \oplus p_3$ can be sent with transmission rate $2k/s$, where destination $d_2$ and $d_3$ will receive and decode their “wanted” packets after $7s$. Obviously, the latter solution is better than the first one, as no packet will miss the deadline.

In this paper, we study a new variant of index coding. By considering the impact of both transmission rate and network coding on the packet reception delay, we design a joint rate selection and network coding (RSNC) scheme for wireless time critical applications, so as to maximize the total benefits obtained by successfully receiving the packets without missing their deadlines. Here, benefit can be defined as the QoS or priority, and in this paper, we mainly set the benefit based on the priority or importance of a packet. The main contributions of our paper can be concluded as follows:

- We propose a graph model, which considers both the heterogenous transmission rates and the deadlines of the packet receptions. Based on the graph model, we mathematically formulate the problem of maximizing the total benefits received by the packets that are successfully received at their destinations without missing deadlines, as an integer programming problem.
- For each packet transmission, we propose a metric based on net benefit to determine the coded packet and the transmission rate. By considering the impact of the transmission rate on both delay and network coding gain, we also design an efficient algorithm to optimize the proposed metric.
- We also consider a special case when the delay constraints for all the packets are the same. We study the pairwise coding solution for this special case, and present a polynomial time algorithm which achieves at least $1 - \frac{1}{e}$ of the optimal pairwise coding solution.
- We compare the performance of the proposed RSNC scheme with some existing algorithms. Simulation results show that the proposed scheme can significantly improve the total benefit obtained by the packets that are received without missing their deadlines.

The rest of the paper is organized as follows. In Section II we define and formulate our problem. The algorithm design for general joint rate selection and network coding scheme is given in Section III. We study the pairwise coding solution for a special case in Section IV. In Section V we present the simulation results. Finally, we conclude the paper in Section VI.

### II. Problem Formulation

In this section, we first give the problem description and its complexity. Then, we introduce an auxiliary graph model, which can be used to design the algorithm. Finally, based on the graph model, we mathematically formulate the proposed problem. To ease understanding, the main notations are listed in Table I.

#### Table I: Main Notations and Their Descriptions

| Symbol | Description |
|--------|-------------|
| $B$    | The size of the packet |
| $D$    | The set of destination nodes |
| $d_i$  | The $i$-th destination in $D$ |
| $H(d_i)$ | The set of available packets at destination $d_i$ |
| $P$    | The set of $n$ packets |
| $p_j$  | The $j$-th packet in $P$ |
| $R(d_i)$ | The set of required packets of $d_i$ |
| $r(s,d_i)$ | The maximum transmission rate on transmission link from $s$ to $d_i$ |
| $T_{i,j}$ | The deadline of packet $p_j$ required at $d_i$ |
| $\alpha_{i,j}$ | The benefit of the packet $p_j$ at $d_i$ |
A. Problem Description

In this paper, we consider the application of network coding in wireless broadcasting/multicasting. Without loss of generality, let $s$ be the source/server node to send a data file to its destination nodes in $D = \{d_1, d_2, \ldots, d_m\}$. Assume that the data file is divided into $n$ packets in $P = \{p_1, p_2, \ldots, p_n\}$, and each packet has the same size $B$. Suppose that initially, each destination node has already stored a subset of packets in its buffer (e.g., side information or received from previous broadcasting) \(^{28}\). Let $H(d_i)$ be the set of available packets at $d_i$, and $R(d_i)$ be the set of required packets by $d_i$, i.e., $R(d_i) \subseteq P, H(d_i) \subseteq P$. For each packet $p_j \in R(d_i)$, let $T_{i,j}$ be the reception deadline of packet $p_j$ at node $d_i$. Let $\alpha_{i,j}$ be the benefit (i.e., profit or importance) of packet $p_j$ at node $d_i$. For example, the “benefit” $\alpha_{i,j}$ can be the profit of the packet $p_j$ obtained by source node $s$ if $p_j$ is timely received at destination $d_i$ by deadline $T_{i,j}$. Suppose that $r(s,d_i)$ is the maximum transmission rate on link $(s,d_i)$, and only if the transmission rate from $s$ to $d_i$ is no more than $r(s,d_i)$, the packet sent from $s$ can be successfully received by $d_i$ \(^{20}\).

Assume that $s$ knows the side information that each destination has such that it can perform network coding operation. Such information can be achieved by using reception reports, as introduced in \(^{12}\). We also assume that node $s$ knows the deadlines of the packet receptions at its receivers. As in COPE \(^{12}\), only XORs coding is performed at the node in our work. In addition, our broadcast channel is that at each time, the source node broadcasts a single message to all the receivers at one single rate, which is different from those in the information-theoretic broadcast channel, where the source node can send different packets to different receivers at different rates.

Our problem is that given the information of $H(d_i)$, $R(d_i)$, $T_{i,j}$, $\alpha_{i,j}$ and $r(s,d_i)$ for all $i,j$, we design the encoding strategy of the packets and select the transmission rate for each propagation, such that the total benefit of the packets that are successfully received at their destinations without missing their deadlines is maximum.

Let $z_{i,j}$ be 1 if packet $p_j$ does not miss its deadline at $d_i$, otherwise, let it be 0, where $p_j \in R(d_i)$. Thus, our objective is to maximize

$$\sum_{d_i \in D} \sum_{p_j \in R(d_i)} \alpha_{i,j} z_{i,j}$$

(1)

In this paper, we refer such a problem of joint Rate Selection and Network Coding (RSNC) for time critical applications as RSNC problem. Note that if $\alpha_{i,j} = 1$ for all $i,j$, the RSNC problem becomes to maximize the number of packets that are received without missing their deadlines, which is the case in our previous conference paper \(^{31}\).

**Lemma 1** The RSNC problem is NP-hard.

**Proof:** We consider a special case of the RSNC problem: $\alpha_{i,j} = 1$, $T_{i,j}$ is the same for all $i,j$, and the maximum transmission rates on all the links are the same. Under the above assumptions, the problem is reduced to NP index coding and the NP hard nature of the problem is well known \(^{17}\), \(^{18}\). Thus, the RSNC problem is also NP-hard.

According to the above lemma, we know that the complexity of finding the optimal solution of RSNC problem is exponential.

B. Graph Model

Although the graph model in \(^{27}\) works well for the case where the transmission rates on all the links are the same and fixed, it cannot be used directly for our RSNC problem. Here, we construct a novel graph model $G(V,E)$, which considers both the transmission rates and the packet reception deadlines.

We define $r_{\min}(s,d_i|p_j) = \frac{B}{r_{i,j}}$ as the minimum transmission rate that can be used to meet the deadline of $p_j \in R(d_i)$ at $d_i$. We add a vertex $v_{i,j}$ in $V(G)$, only if the following two conditions can be met.

(1) $p_j \in R(d_i)$;
(2) $r_{\min}(s,d_i|p_j) \leq r(s,d_i)$.

Note that, if $r_{\min}(s,d_i|p_j) > r(s,d_i)$, packet $p_j$ will definitely miss its deadline at $d_i$. Thus, conditions (1) and (2) ensure that we add a vertex $v_{i,j}$ in $V(G)$ only if the “wanted” packet $p_j$ may not miss its deadline at $d_i$. That is, $V(G) = \{v_{i,j}|p_j \in R(d_i), r_{\min}(s,d_i|p_j) \leq r(s,d_i)\}$.

Then, for any two different vertices $v_{i,j}, v'_{i,j} \in E(G)$, there is an edge $(v_{i,j},v'_{i,j}) \in E(G)$ if all the following conditions can be satisfied:

(a) $i \neq i'$;
(b) $j = j'$ or $p_j \in H(d_i')$ and $p_{j'} \in H(d_i)$;
(c) $r_{\min}(s,d_i|p_j) \leq r(s,d_i')$ and $r_{\min}(s,d_i'|p_{j'}) \leq r(s,d_i)$.

We also define the weight of vertex $v_{i,j}$ as the benefit of packet $p_j$ at $d_i$, $\alpha_{i,j}$. For any clique $Q = \{v_{i_1,j_1}, v_{i_2,j_2}, \ldots\}$ in $G$, let $P' = \{p_j|v_{i,j} \in Q\}, D' = \{d_i|v_{i,j} \in Q\}$.

Similar to the work in \(^{27}\), if node $d_i \in D'$ successfully receives the encoded packet $p_j \oplus p_{j_2} \oplus \ldots \oplus p_{j_t}$, where $p_{j_1}, p_{j_2}, \ldots, p_{j_t} \in P'$, $d_i$ can decode a “wanted” packet $p_j$, where $v_{i,j} \in Q$.

Next, we will use an example to show the novelty of our graph model as compared to others in the literature, e.g., \(^{27}\). Still take Fig. 1 as an example. The graph constructed by \(^{27}\) is shown in Fig. 2 (a). According to \(^{27}\), any clique in the graph represents a feasible encoded packet. Thus, $p_1 \oplus p_2 \oplus p_3$ can be sent and its intended next hops are $d_1, d_2, d_3$, because $\{v_{1,1}, v_{2,2}, v_{3,3}\}$ forms a clique. As described before, it is not a good choice, as we cannot find a transmission rate to meet the deadlines of all the packets. However, with our graph model shown in Fig. 2 (b), $p_1 \oplus p_2 \oplus p_3$ will not be encoded because vertices $v_{1,1}, v_{2,2}, v_{3,3}$ do not form a clique in the graph. In addition, for the current transmission, the encoded packet derived from any clique in the graph can be sent without missing the deadlines at its intended destinations. For example, if $p_2 \oplus p_3$, which is derived from the clique $\{v_{2,2}, v_{3,3}\}$, is sent with the minimum of the possible transmission rates among $r(s,d_2)$ and $r(s,d_3)$, $2k/s$, its intended next hops $d_2, d_3$ can successfully decode the packets $p_2, p_3$ respectively without missing their deadlines.

For any clique $Q$ in the graph $G(V,E)$, we have the following lemma.
Lemma 2 In the current transmission, if the encoded packet $p_{j_1} \oplus p_{j_2} \oplus \cdots \oplus p_{j_{|P'|}}$, where $p_{j_1}, \cdots, p_{j_{|P'|}} \in P'$ and $P' = \{ p_i | v_i \in Q \}$, is sent with the transmission rate $r = \min \{ r(s, d_i) | d_i \in D' \}$, it will be received by all the nodes in $D'$. Then, for each $v_{i,j} \in Q$, the packet $p_j$ will be decoded by $d_i$ without missing its deadline. In addition, the benefit obtained by such a transmission is the sum of the weights on the vertices in the clique, i.e., $\sum_{v_{i,j} \in Q} \alpha_{i,j}$.

Proof: Firstly, we can easily obtain that with transmission rate $r = \min \{ r(s, d_i) | d_i \in D' \}$, all the receivers in $D'$ can successfully receive the sending packet. This is because the transmission rate $r$ must be lower than the maximum transmission rate from $s$ to any $d_i \in D'$.

Secondly, the constructed graph satisfies the decoding property given in [27]. According to [27], if $p_{j_1} \oplus p_{j_2} \oplus \cdots \oplus p_{j_{|P'|}}$ is successfully received by $d_i \in D'$, $d_i$ can decode its “wanted” packet $p_j$, where $v_{i,j} \in Q$. Thus, any receiver $d_i \in D'$ can obtain a “wanted” packet $p_j$ from $p_{j_1} \oplus p_{j_2} \oplus \cdots \oplus p_{j_{|P'|}}$ with transmission rate $r$, where $v_{i,j} \in Q$.

Thirdly, according to the condition (c), we have

$$r_{\min}(s, d_i | p_j) \leq \min_{d_i \in D'} \{ r(s, d_i) \} = r$$

(2)

So, its arrival time at receiver $d_i$ is

$$\frac{B}{r} \leq r_{\min}(s, d_i | p_j) = \frac{B}{r_{i,j}} = T_{i,j}$$

(3)

In other words, the arrival time of the packet $p_j \in P'$ will not miss its deadline at its receiver $d_i \in D'$, where $v_{i,j} \in Q$.

With Lemma 2, a clique $Q$ in the graph represents a feasible transmission solution for the current propagation, with the encoded packet $p_{j_1} \oplus p_{j_2} \oplus \cdots \oplus p_{j_{|P'|}}$, transmission rate $r = \min \{ r(s, d_i) | d_i \in D' \}$, the propagation delay $\frac{B}{r}$, and the benefit $\sum_{v_{i,j} \in Q} \alpha_{i,j}$.

C. RSNC Formulation

While Lemma 2 ensures that any encoding strategy based on any clique in the graph will be delivered within deadline for the current packet transmission, the transmission orders of the encoded packets, represented by the cliques in $G(V, E)$, is important for the timely packet receptions at their destinations.

For example, as shown in Fig. 2(b) where we assume that $\alpha_{i,j} = 1$ for all $i, j$, if we first schedule packet $p_1$ with transmission rate $5k/s$, represented by clique $\{v_{1,1}\}$, and then schedule packet $p_2 \oplus p_3$ with transmission rate $2k/s$, represented by clique $\{v_{2,2}, v_{3,3}\}$, all the packets will be received at their destinations without missing deadlines. The total benefit with such a solution is $\alpha_{1,1} + \alpha_{2,2} + \alpha_{3,3} = 3$.

However, if we first schedule packet $p_2 \oplus p_3$, and then packet $p_1$, packet $p_1$ will miss its deadline at $d_1$. Correspondingly, the total benefit with the latter solution is only $\alpha_{2,2} + \alpha_{3,3} = 2$.

Thus, our next task is to find a set of cliques in the graph and schedule the transmissions of the encoded packets represented by these cliques, so as to maximize the total benefit of the packets that are successfully received/decoded at their destinations without missing their deadlines. Suppose that $Q_h = \{ v_{i_1,j_1}, v_{i_2,j_2}, \cdots \}$ is the $h$-th clique found in the graph, and the corresponding encoded packet represented by $Q_h$ is sent as the $h$-th transmission at node $s$. We also assume that $P'_h = \{ p_j | v_{i,j} \in Q_h \}$, $D'_h = \{ d_i | v_{i,j} \in Q_h \}$. Thus, the packet sent by the $h$-th transmission at $s$ is $p_{j_1} \oplus p_{j_2} \oplus \cdots \oplus p_{j_{|P'_h|}}$, where $p_{j_1}, p_{j_2}, \cdots, p_{j_{|P'_h|}} \in P'_h$, and the transmission rate is $r_h = \min_{d_i \in D'_h} \{ r(s, d_i) \}$. Let $T_h$ be the transmission delay of the $h$-th transmission, i.e., $T_h = \frac{B}{r_h}$.

We first define the following variant.

$$x_{i,j,h} = \begin{cases} 1, & \text{if vertex } v_{i,j} \text{ is included in clique } Q_h \\ 0, & \text{otherwise} \end{cases}$$

(4)

Then, we can formulate the RSNC problem based on the graph model as follows.

$$\max_{\{ Q_h \}} \sum_{d_i \in D} \sum_{p_j \in R(d_i)} \alpha_{i,j} x_{i,j}$$

subject to

$$\sum_{h=1}^{\lvert V(G) \rvert} x_{i,j,h} = 1, \forall v_{i,j} \in V(G)$$

(6)

$$x_{i,j,h} + x_{i',j',h} = 1, \forall h \in \{ 1, 2, \cdots, |V(G)| \}, i, j \neq i', j', \forall v_{i,j}, v_{i',j'} \notin E(G)$$

(7)

$$T'_h = \max_{v_{i,j} \in V} \{ B \cdot x_{i,j,h} / \{ r(s, d_i) \} \}, 1 \leq h \leq |V(G)|$$

(8)

$$\sum_{h=1}^{\lvert V(G) \rvert} (x_{i,j,h} \ast \sum_{j=1}^{h} T'_j) \leq T_{i,j} + \xi z_{i,j}, \forall v_{i,j}$$

(9)

$$\sum_{h=1}^{\lvert V(G) \rvert} (x_{i,j,h} \ast \sum_{j=1}^{h} T'_j) \geq T_{i,j} - \xi (1 - z_{i,j}), \forall v_{i,j}$$

(10)

$$z_{i,j} \in \{ 0, 1 \}, \forall v_{i,j}$$

(11)

where $\xi$ is a sufficient large constant.

In the above formulation, the term of the objective represents the total benefit of the packets that are successfully received at their destinations without missing their deadlines, which needs to be maximized. Constraint (6) denotes that each vertex in the graph can only belong to one clique. Constraint (7) means that if there is no edge between vertex $v_{i,j}$ and $v_{i',j'}$, vertices $v_{i,j}$ and $v_{i',j'}$ cannot be in the same clique. Constraint (8) gives the transmission delay for the $h$-th transmission, which is equal to the transmission delay with the minimum transmission rate among the rates from $s$ to all intended receivers. The sufficient large constant $\xi$ is used to guarantee that if $\sum_{h=1}^{\lvert V \rvert} (x_{i,j,h} \ast \sum_{j=1}^{h} T'_j) > T_{i,j}$, $z_{i,j}$ must be 1, as denoted in Constraint (10), and if
maximize the metric $U$. In other words, we would like to propose a greedy solution to optimize the current performance only.

For the current transmission, given encoded packet $p_{j_1} \oplus p_{j_2} \oplus \cdots \oplus p_{j_L}$ and the transmission rate $r$, $f_{i,j}$ and $l_{i,j}$ can be both determined. For example, $f_{i,j}$ is 1 if and only if all the following conditions are met:

- $r \geq r(s, d_i)$, which means $d_i$ can successfully receive the sending packet;
- $p_j \in R(d_i)$, which means $p_j$ is required by $d_i$;
- All the other packets encoded in the current packet except $p_j$ are available at $d_i$, which is the decoding requirement of $p_j$ at $d_i$;
- $\frac{B}{r} \leq T_{i,j}$, which shows the requirement of the reception deadline.

In addition, $l_{i,j}$ is 1 if and only if for $\forall p_j \in R(d_i)$,

$$\frac{B}{r} + \frac{B}{r(s, d_i)} > T_{i,j} \tag{14}$$

Here, $\frac{B}{r}$ is the transmission delay of the current transmission, and $\frac{B}{r(s, d_i)}$ denotes the minimum delay to meet $p_j$’s deadline at $d_i$ in the next transmission. If the sum of the current transmission delay and the next minimum transmission delay is larger than the deadline of $p_j$ at $d_i$, $p_j$ will definitely miss its deadline, i.e., $l_{i,j} = 1$.

Note that, the problem of maximizing the defined metric $U$ is also NP-hard. We can prove it by considering its special case: the transmission rates on all the links are the same, the reception deadlines for all the packets are the transmission time of one packet, and each packet has the same benefit $\alpha_{i,j}$. The special case of maximizing the defined metric $U$ becomes to maximize the total number of the receivers that can decode one “wanted” packet from the current encoded packet, which has been proved to be NP-hard in [22].

B. Heuristic Algorithm Design for Each Packet Propagation

Although maximizing the defined metric $U$ is NP-hard, we can easily obtain the following observations, based on which we can design a heuristic algorithm.

1: Maximizing the first term of the metric $U$ is equal to find a maximum weight clique in the graph, where the weight at vertex $v_{i,j}$ is defined as the benefit $\alpha_{i,j}$.

2: The transmission rate is a parameter that adjusts the trade-off between delay and network coding gain. If $s$ uses a low transmission rate, more receivers can successfully receive the sending packet, and the current transmission may satisfy more receivers’ requirements, denoted by the first term in $U$. However, low transmission rate means high transmission delay, which may cause more packets to miss their deadlines in the following transmissions, denoted by the second term in $U$.

Based on the above observations, we then design a heuristic algorithm for each packet propagation, by gradually increasing the transmission rate. Initially, the transmission rate is set to be no less than the lowest one from $s$ to its receivers. Let $TR = \{r(s, d_i) | d_i \in D\}$ be the set of available transmission rates from $s$ to all the destinations, and let $T_{rk}$ be the $k$-th lowest rate in $TR$. As in Section III-B, we construct the auxiliary graph with the given information.
In the $k$-th step, we restrict that the transmission rate used at $s$ must be no less than $T_{R,k}$. For $d_i$, if its maximum transmission rate from $s$ is less than $T_{R,k}$, it cannot successfully receive the sending packet. This restriction can be realized by omitting any vertex $v_{i,j}$ in $G(V,E)$ if $r(s,d_i) < T_{R,k}$. Then, we find the maximum weight clique in the subgraph \{$v_{i,j} | r(s,d_i) \geq T_{R,k}, v_{i,j} \in V(G)$\}, and adopt the transmission rate represented by the found clique. Each vertex $v_{i,j}$ in the found clique denotes that $p_j$ will be successfully received/decoded by $d_i$ without missing its deadline, for the given transmission rate. For each of the other packets that cannot be obtained at their receivers from the current transmission, we then judge whether it will definitely miss its deadline at its destinations, by \[14\]. Thus, in each step, we calculate $U$. Such process continues until all the rates in $TR$ are considered. Finally, we compare the values of $U$ obtained from each step and adopt the one with the largest value as the solution. Note that, if there are more than one solution with the maximum value of $U$, we will choose the one with the smaller lost represented by the second term of \[13\]. The detailed of the algorithm is shown in Algorithm \[11\].

C. Algorithm for the Whole Transmission Process

While Algorithm 1 in Section \[11-B\] describes the encoding strategy of the packets and the selection of the transmission rate for every propagation, the whole transmission process will consist of multiple of such single process. We will first construct the graph $G(V,E)$ based on the model in Section \[11-B\] and the graph will be updated by removing the selected vertices in the found clique by Algorithm \[11\] and the vertex $v_{i,j}$ if $p_j$ will definitely miss its deadline at destination $d_i$. The packet reception deadlines for the packets also need to be updated after each transmission. The whole transmission process continues until the vertices set $V$ of $G$ becomes empty. The detail algorithm for the whole transmission is given in Algorithm \[12\].

D. Algorithm Complexity

In this section, we will analyze the complexity of the proposed algorithm.

We first discuss the complexity of the algorithm for single packet propagation. According to Algorithm 1 for each available transmission rate, the encoding strategy and rate selection is converted into finding a maximum weight clique in the defined subgraph. As finding a maximum weight clique in the graph is also an NP-hard problem, in this paper, we exploit a heuristic algorithm \[32\], whose complexity is $O(|V(G)|^3)$. By trying all the available transmission rates in $TR$, the complexity of determining single packet propagation is $O(|V(G)|^3|TR|)$. As the maximum number of transmissions is at least the number of packets in $P$, the complexity of the algorithm for the whole transmission process is $O(|V(G)|^3|TR||P|)$. In other words, the complexity of the above proposed algorithm is $O(|V(G)|^3|TR||P|)$.

IV. PAIRWISE CODING FOR THE SAME DELAY CONSTRAINT

We now consider a special case when the delay constraints for all the packets are the same, e.g. in a video broadcasting, all packets have the same delay constraints to all the clients. As in \[20, 33\], we focus on a practically coding scheme, pairwise coding (or sparse coding in \[33\]), so as to decrease the encoding and decoding complexity. In this section, we first give a brief review on pairwise coding. Then, we design a greedy algorithm, and prove that the proposed algorithm achieves at least $(1 - \frac{1}{l})$ of the optimal solution to pairwise
coding, where $e$ is the base of the natural logarithm.

A. Pairwise Coding

With pairwise coding [20], [33], at most two requests of the destinations can be satisfied within one transmission, i.e., at most two packets are encoded together. As in Section II-B, each clique in the graph model $G(V, E)$ represents a feasible packet transmission. In other words, with pairwise coding, the size of the clique selected for each transmission is at most two.

Without loss of generality, let $T$ be the delay constraint for all the packets. Based on the graph $G(V, E)$ constructed in Section II-B, we can enumerate all the cliques with at most $Q$.

Let $\alpha_{i,j}$ be the set of all the cliques defined above. For each element (vertex) included in the clique, we also define $\alpha_{i,j}$ as its weight. Then, the weight of the clique $Q_h \in Q$ is denoted as $w_h$, can be defined as the sum of the weights on the elements covered by it, i.e., $w_h = \sum_{\alpha_{i,j} \subseteq Q_h} \alpha_{i,j}$. Note that in the following, the term “weight” has the same meaning as the term “benefit”. For each clique $Q_h \in Q$, we also define the cost $c_h$ as the maximum transmission delay to satisfy the requests represented by the vertices in $Q_h$, i.e.,

$$c_h = \max_{\alpha_{i,j} \subseteq Q_h} \left( \frac{B}{R} \right).$$

With the above definition, the cost of clique $Q_h$ denotes the transmission delay of sending the encoded packet represented by $Q_h$, and the weight of clique $Q_h$ denotes the benefit of scheduling the encoded packet represented by $Q_h$ before deadline $T$. Then, pairwise coding to maximize the total benefit is converted into finding a collection of cliques in $Q^* \subseteq Q$ such that total weight of all the selected cliques in $Q^*$ is maximum, while the following conditions need to be satisfied:

1. The total cost of the cliques in $Q^*$ does not exceed a given budget $T$, i.e., $\sum_{Q_h \in Q} c_h \leq T$, which denotes the delay constraint;

2. For each two cliques $Q_h$ and $Q_{h'}$ in $Q^*$, they must satisfy $Q_h \cap Q_{h'} = \emptyset$, which means that the satisfied request will be excluded in the following transmissions.

To ease understanding, we can formulate the above problem as follows.

$$\max_{Q^* \subseteq Q} \sum_{Q_h \in Q^*} w_h$$

subject to

$$\sum_{Q_h \in Q^*} c_h \leq T, \quad Q_h \cap Q_{h'} = \emptyset, \forall Q_h, Q_{h'} \in Q^*$$

We can easily prove that the above pairwise coding problem is NP hard, by reduction a classical NP problem, 0-1 knapsack problem, to a special case of the proposed problem when there is no edge in graph $G(V, E)$.

B. Algorithm Design

We design a greedy algorithm to realize the pairwise coding, which is conducted with iterations. To facilitate the further discussion, we define the following parameters.

- $Q_k^*$: the collection of cliques selected in the first $k$ iterations, where $Q^*_k \subseteq Q$.
- $C(Q_k^*)$: the sum of the cost for the cliques in $Q_k^*$.
- $W(Q_k^*)$: the total weight of the elements covered by the cliques in $Q_k^*$.
- $w_k(Q_h)$: the total weight of the elements covered by clique $Q_h$, but not covered by any clique in $Q_k^*$.

Without loss of generality, we assume that the cost of each clique in $Q$ is no more than $T$, since the cliques whose costs are more than $T$ must not belong to any feasible solution.

Initially, suppose that $U^* = Q$ and $Q_0 = \emptyset$. Our algorithm is conducted with iterations.

- In the $k$-th iteration, we consider the clique $Q_h$ in $U^*$ that maximizes the ratio $w_k(Q_h)$.
- If adding cost $c_h$ to $C(Q_k^*)$ is more than $T$, discard the clique $Q_h$ from $U^*$.
- If adding cost $c_h$ to $C(Q_k^*)$ is less than $T$, we then check if $Q_h$ has intersection with the selected cliques in $Q_{k-1}^*$. If there is no intersection, add the clique $Q_h$ to the found collections, i.e., $Q_{k}^* = Q_{k-1}^* \cup \{Q_h\}$. Correspondingly, we delete $Q_h$ from $U^*$.
- If there is intersection but the intersection set is not empty, there must exist another clique $Q_{h'}$ left in $U^*$, which includes the elements covered by clique $Q_h$ but not covered by any clique in $Q_{k-1}^*$, i.e., $Q_{h'} = Q_h - Q_{k-1}^* \cap \{v_{i,j} \mid v_{i,j} \in Q_h \}$ and $Q_{h'} \in Q_{k-1}^*$. We then add clique $Q_{h'}$ to the selected collection, i.e., $Q_{k}^* = Q_{k-1}^* \cup Q_{h'}$, and delete both $Q_h$ and $Q_{h'}$ from $U^*$.
- If there is intersection but the intersection set is empty, we just delete $Q_h$ from $U^*$.
- The above process continues until $U^* = \emptyset$.

Finally, we compare the total weight of the elements covered by the selected cliques so far and the maximum weight of the clique in $Q$, and keep the maximum one as the solution. The detail algorithm is shown in Algorithm 3.

C. Performance Analysis

Before analyzing the performance of our algorithm, we first give a brief review on a classic budgeted set coverage problem [34], which is similar to our problem and will be used in the following analysis.

1) Budgeted Set Coverage Problem: The objective of the budgeted set coverage problem is to find a collection of the sets in $Q$ such that the total weight of the elements covered by the selected sets is maximum while the total cost of all the selected sets is no more than a given value $T$. Compared with our problem, the budgeted set coverage problem does not require that the selected sets (i.e., cliques in our case) should be disjoint with each other. In the following presentation, we use the cliques and sets to denote the same meaning.

The algorithm in [34] is conducted with iterations. Let $Q_k' \subseteq Q$ be a collection of the cliques selected in the first $k$ iterations.
Algorithm 3: Algorithm design for pairwise coding

\begin{algorithm}
\begin{algorithmic}
\State $Q_k^* \leftarrow \emptyset, C(Q_k^*) = 0, w_h(Q_h) = 0, U^* \leftarrow Q$, for \(\forall k, h\);
\State $k = 0$;
\While{$U^* \neq \emptyset$}
\State Select $Q_h \in U^*$ that maximizes $\frac{w_{k-1}(Q_h)}{c_h}$;
\If{$C(Q_{k-1}) + c_h > T$}
\State $Q_k^* \leftarrow Q_{k-1};$
\State $C(Q_k^*) = C(Q_{k-1}) + c_h$;
\Else
\If{there is no interaction between $Q_h$ and cliques in $Q_k^*$}
\State $Q_k^* \leftarrow Q_{k-1} \cup \{Q_h\}$;
\State $C(Q_k^*) = C(Q_{k-1}) + c_h$;
\Else
\If{there exists another clique $Q_{h'} \in U^*$ that covers elements in $Q_h$, but not in any clique in $Q_k^*$}
\State $Q_k^* = Q_{k-1} \cup \{Q_{h'}\}$;
\State $C(Q_k^*) = C(Q_{k-1}) + c_{h'}$;
\EndIf
\EndIf
\State $U^* \leftarrow U^* - Q_h$;
\EndIf
\EndWhile
\State Consider the clique $Q_h$ that maximize $w_h$ over $Q$;
\If{$W(Q_h) \geq w_h$, output $Q_k^*$, otherwise, output $\{Q_h\}$;
\end{algorithmic}
\end{algorithm}

Let $w_h'(Q_h)$ denote the total weight of the elements covered by $Q_h$, but not covered by any clique in $Q_h^\prime$. Initially, suppose that $U' = Q$. The main idea of the algorithm in [34] is that in each iteration, the unconsidered clique $Q_h \in U'$ that maximizes the ratio of weight $w_{k-1}'(Q_h)$ to its cost (i.e., $\frac{w_{k-1}'(Q_h)}{c_h}$) is considered. If adding $c_h$ exceeds $T$, discard $Q_h$, otherwise, add $Q_h$ to the collection of selected cliques so far. Also, delete the considered clique $Q_h$ from $U'$. The above operation continues until all the cliques in $U'$ are considered.

Although the algorithm in [34] conducts well for the budgeted set coverage problem, it cannot be used to solve our problem. This is because our problem requires the elements included in each pair of cliques in $Q'$ to be disjoint with each other, as denoted in Constraint (2), which differentiates from the budgeted set coverage problem.

2) Analysis of the Proposed Pairwise Coding Solution:

Without loss of generality, let $Q^P_j \in Q'$ and $Q^j_{j'}$ be the $t$-th selected cliques with algorithm in [34] and our Algorithm [8], respectively. Let $l^j_q$ and $l^j_{q'}$ be the index of the iteration in which clique $Q^j_{j'}$ and clique $Q^j_q$ are selected, respectively. We assume that $Q' = \{Q^j_1, Q^j_2, \ldots, Q^j_q\}$ is the collection of all the selected cliques with algorithm in [34], and $Q^* = \{Q^j_1, Q^j_2, \ldots, Q^j_q, \ldots, Q^*_k\}$ is the collection of all the selected cliques with Algorithm [8]. Correspondingly, we let $S'_k$ and $S^*_k$ be the set of elements covered by the cliques in $Q_k'$ and $Q_k^*$ respectively.

Without loss of generality, we assume that each clique $Q^j_q \in Q'$ (or $Q^j_q \in Q^*$) satisfies that $w_{t-1}'(Q^j_q) > 0$ (or $w_{t-1}^*(Q^j_q) > 0$), otherwise, adding it has no contribution in increasing the total benefit (i.e., weight) and thus does not need to be considered. Let $Q^P_q$ and $Q^P_q^*$ be the original cliques considered in the $k$-th iteration whose ratio $\frac{w_{t-1}'(Q^j_q)}{c^j_q}$ are the maximum among all the unconsidered cliques so far in $U'$ and $U^*$ with algorithm in [34] and Algorithm [8], respectively.

By comparing $Q^P_q$ and $Q^P_q^*$, we can obtain the following lemma.

Lemma 3 At the $k$-th iteration, the clique $Q^P_q$ considered by the algorithm in [34] is the same as the clique $Q^P_q^*$ considered by our Algorithm [8], i.e., $Q^P_q = Q^P_q^*$, where $1 \leq k \leq \min\{t'_q, t^*_q\}$.

**Proof:** See Appendix A

We can also obtain that

Lemma 4 At each $k$-th iteration, if the algorithm in [34] selects (does not select) a clique, the Algorithm [8] must also select (not select) a clique, where $1 \leq k \leq \min\{t'_q, t^*_q\}$.

**Proof:** See Appendix B

According to Lemma 3 and Lemma 4, we can get the following Lemma.

Lemma 5 After the $k$-th iteration, the cliques in the collections $Q_k'$ and $Q_k^*$ cover the same set of the elements, i.e., $S'_k = S^*_k$, where $1 \leq k \leq \min\{t'_q, t^*_q\}$.

**Proof:** See Appendix C

Based on the above lemmas, we can further obtain the following lemma.

Lemma 6 The number of cliques added in $Q'$ must be no more than the number of cliques added in $Q^*$, i.e., $q' \leq q^*$, and $t_q \leq l_q^*$.

**Proof:** See Appendix D

With the above lemmas, we can compare the weights of the elements covered by the two algorithms as follows.

Lemma 7 The total weight of the elements covered by $Q'$ must be less than the total weight of the elements covered by $Q^*$, i.e., $W(Q') \leq W(Q^*)$.

**Proof:** See Appendix E

Based on the above results, we then prove the approximation ratio as follows.

Theorem 1 The total benefit achieved with our Algorithm [8] is at least $1 - \frac{1}{e}$ of the optimal solution with pairwise coding.

**Proof:** See Appendix F
V. Simulation Results

In this section, we demonstrate the effectiveness of our RSNC scheme through simulations. We randomly generate a set of available packets in \( H(d_i) \) and the “wanted” packets in \( R(d_i) \) at destination \( d_i \in D \), where \( H(d_i) \cap R(d_i) = \emptyset \). The maximum transmission rate from \( s \) to \( d_i \) is randomly selected in \([r_{\text{min}}, r_{\text{max}}]\), and the packet reception deadline is randomly generated in \([T_{\text{min}}, T_{\text{max}}]\).

For comparison purpose, we mainly include two baseline algorithms, namely, DSF (deadline smallest first) coding algorithm [27] and SIN-1 algorithm [1]. DSF coding algorithm does not consider the heterogenous transmission rates on the links, and in each time slot, it always finds the maximum weight clique in the defined graph. SIN-1 algorithm always sends the packet with the minimum “SIN-1” in each transmission, where “SIN-1” of packet \( p_j \) is defined as the ratio of the most urgent deadline of the requests for \( p_j \) to the total number of requests for \( p_j \). We also include Random Linear Network Coding and Index Coding as baseline algorithms in Section V-F.

In the simulation, we compare the total benefit obtained by timely receiving the required packets at their destinations, under different transmission schemes. For each setting, we present the average result of 200 samples.

A. The Impact of the Transmission Rate

We first investigate the impact of the transmission rate on the performance of the designed Algorithm 2 for the whole transmission process. We randomly generate the benefit of each packet (i.e., \( \alpha_{i,j} \)) from 0.5 to 2. We also set \( n = m = 10, T_{\text{min}} = 10, T_{\text{max}} = 50 \) and vary the scale of the transmission rates, i.e., \([r_{\text{min}}, r_{\text{max}}]\).

As shown in Fig. 3 with our RSNC scheme, the total benefit obtained by timely receiving the required packets at the destinations is much higher than the other two schemes. This is because RSNC not only considers the packet reception deadline for each packet, but also utilizes the heterogenous transmission rates from \( s \) to the receiver nodes. We also can see that with the increase of the transmission rates, the total benefit increases. The reason is that higher transmission rates incur less transmission delay, which can satisfy more timely receptions in the following transmissions.

B. The Impact of the Number of Destinations \( m \)

We then investigate the impact of the number of destinations \( m \) and the transmission rates on the deadline miss ratio. The benefit of each packet (i.e., \( \alpha_{i,j} \)) is randomly generated from 0.5 to 2. We set \( n = 10, T_{\text{min}} = 10, T_{\text{max}} = 50 \) by varying \( m \) in \([5, 15]\) for \( r_{\text{min}} = 10, r_{\text{max}} = 50 \) and \( r_{\text{min}} = 50, r_{\text{max}} = 100 \).

As shown in Fig. 4 with our RSNC scheme, the total weight obtained by timely receiving the required packets at the destinations, is much more than with other schemes. We can also see that the DSF algorithm does not show significant gain over SIN-1 algorithm. This is because, although with network coding in DSF, more packets can be combined together, the encoded packet may still miss its deadline at some destinations, due to inappropriate transmission rate used.

In addition, with the increase of \( m \), the total obtained benefit increases, as more requests need to be satisfied, which gives higher chance to obtain more benefit. By comparing Fig. 4(a) and (b), we observe that, with the increase of the transmission rates, the total benefit increases, similar to Fig. 3.

C. The Impact of the Number of Packets \( n \)

We now investigate the impact of the total number of packets \( n \) and the reception deadlines on the total benefit obtained by timely receiving the required packets at the receiver nodes. The benefit of each packet (i.e., \( \alpha_{i,j} \)) is randomly generated from 0.5 to 2. We set \( m = 10, r_{\text{min}} = 10, r_{\text{max}} = 50 \) by varying \( n \) in \([10, 40]\) for the cases of \( T_{\text{min}} = 10, T_{\text{max}} = 50 \) and \( T_{\text{min}} = 10, T_{\text{max}} = 80 \).

From Fig. 5 we can see that our proposed RSNC scheme achieves the largest benefit while at the same time ensure timely receiving the required packets. In addition, with the increase of \( n \), the total benefit increases. This is because more requests need to be satisfied at node \( s \), which thus has higher chance to get more benefit. From Fig. 5 it is easy to see that the total benefit with \( T_{\text{max}} = 80 \) is smaller than with \( T_{\text{max}} = 50 \). It is reasonable because with the increase of the deadlines, less packet will lose its deadline at its destination node.
D. The Deadline Miss Ratio

We now investigate the performance of the deadline miss ratio, which is defined as the ratio of the number of the requests that miss their deadlines at the destinations, to the total number of the requests. In this case, we set every packet with the same benefit $\alpha_{i,j} = 1$, so the total benefit directly relates to the total number of packets that are successfully delivered within the deadline. We also set $n = 10, T_{min} = 10, T_{max} = 50$ by varying $m$ in $[5, 15]$ for $r_{min} = 10, r_{max} = 50$ and $r_{min} = 50, r_{max} = 100$.

As shown in Fig. 6, the deadline miss ratio with our RSNC scheme is much lower than with other schemes. With the increase of $m$, the gain of our RSNC scheme increases. We can also see that the DSF algorithm does not show significant gain over SIN-1 algorithm. This is because, although with network coding in DSF, more packets can be combined together, the encoded packet may still miss its deadline at some destinations, due to inappropriate transmission rate used. From Fig. 6 we see that, with the increase of $m$, the deadline miss ratio increases. The reason is that there are more packets to be sent at $s$ within the same deadline scale.

E. The Impact of $\alpha_{i,j}$

We now study the impact of the benefit $\alpha_{i,j}$ on the performance of the proposed algorithm. To fairly comparison, the total number of requests are set to be 40, and each request has the same delay constraint. In addition, we set two kinds of benefits: $\alpha_A$ and $\alpha_B$, where the first 20 requests are with benefit $\alpha_A$, while the other 20 requests are with benefit $\alpha_B$. We also set $\alpha_A = 1$ and vary $\alpha_B$ between $[1, 5]$.

To study the impact of the benefits on the performance of the proposed algorithm, we define the successful ratio for requests with higher benefit as the ratio of the number of requests that are with benefit $\alpha_B$ and timely received at the destinations, to the total number of requests that are timely received at the destinations. As shown in Fig. 7 when $\alpha_A = \alpha_B = 1$, the successful ratio for requests with higher benefit is almost 0.5. This is because, with the same benefit/priority, the number of successful requests with $\alpha_A$ is almost the same as the number of successful requests with $\alpha_B$. We can also see that, with the increase of $\alpha_B$, the successful ratio for requests with higher benefit increases significantly, as the requests with higher benefit have higher priority to be scheduled.

F. Comparison with Random Linear Network Coding and Index Coding

To compare the performance of the proposed RSNC algorithm with random linear network coding and index coding, we investigate the performance of the deadline miss ratio. We set $n = 10, T_{min} = 10, T_{max} = 50$ by varying $m$ in $[5, 15]$ for $r_{min} = 10, r_{max} = 100$.

As shown in Fig. 8 the deadline miss ratio with the proposed RSNC scheme is much lower than that with random linear network coding and index coding. This is because either random linear network coding or index coding considers the deadline of packet reception. Particularly, with random linear
network coding, the destination node cannot decode the native packet until receiving the full-rank encoded packets, which further delay the reception/decoding of the native packets. We can also see that with the increase of \( m \), the gain of our RSNC scheme increases.

G. Pairwise Coding

In this subsection, we investigate the performance of pairwise coding for a special case when the packet reception deadline is the same for each packet. We compare the proposed Algorithm 3 with the optimal pairwise coding obtained with ILP by Eq. (15) and Eq. (16). In this simulation, we set \( m = 10 \), \( n = 10 \), \( B = 100k \), \( r_{min} = 10k/s \), \( r_{max} = 100k/s \). The benefit for timely receiving packet \( p_j \) is randomly selected from 1 to 10. We conduct the simulation by varying the packet reception deadline \( T \) in [5, 35].

As shown in Fig. 9, the total benefit obtained with our greedy Algorithm 3 is more than \( (1 - \frac{1}{m}) \) of optimal solution to pairwise coding, which verifies our analysis result. We also can find that the total benefit achieved with our algorithm increases with the increase of the packet reception deadline. This is reasonable since more packets can be timely scheduled before missing the deadline, when the packet reception deadline increases.

VI. CONCLUSION

In this paper, we propose a novel joint rate selection and network coding (RSNC) scheme for time critical applications. We first prove that the proposed problem is NP-hard, and design a novel graph model to model the problem. Using the graph model, we mathematically formulate the problem. We also propose a metric, based on which we design a heuristic algorithm to determine transmission rate and coding strategy for each transmission so as to maximize the total benefit by timely receiving the packets. The benefit can be defined based on the priority or importance of a packet. We then study the pairwise coding for a special case when all the deadlines for all the packets are the same, and design an efficient algorithm which can achieve at least \( 1 - \frac{1}{m} \) of the optimal solution for pairwise coding. Finally, simulation results demonstrate the proposed RSNC algorithm effectively increases the benefit and ensures timely packet receptions at the destination nodes.

REFERENCES

[1] J. Xu, X. Tang, and W. Lee, “Time-critical on-demand data broadcast: algorithms, analysis, and performance evaluation,” IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 1, pp. 3 – 14, 2006.
[2] S. Acharya and S. Muthukrishnan, “Scheduling on-demand broadcasters: new metrics and algorithms,” in Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking. New York, NY, USA: ACM, 1998.
[3] H. Dykeman, M. Ammar, and J. Wong, “Scheduling algorithms for videotex systems under broadcast delivery,” in IEEE International Conference on Communications (ICC), Oct. 1986, pp. 1 – 8.
[4] D. Aksoy and M. Franklin, “\( R \times w \) a scheduling approach for large-scale on-demand data broadcast,” IEEE/ACM Transactions on Networking, Dec. 1999, pp. 846 – 860.
[5] P. Xuan, S. Sen, O. Gonzalez, J. Fernandez, and K. Ramamritham, “Broadcast on demand: efficient and timely dissemination of data in mobile environments,” in Proceedings of the Third IEEE Real-Time Technology and Applications Symposium, Jun. 1997, pp. 38 – 48.
[6] L. Dai, Z. Wang, J. Wang, and Z. Yang, “out:ionizing with OFDM Signals for the Next-Generation GNSS,” in IEEE Transactions on Consumer Electronics, May 2010, pp. 374 – 379.
[7] S. Li, R. Yeung, and N. Cai, “Linear network coding,” Information Theory, IEEE Transactions on, vol. 49, no. 2, pp. 371 – 381, Feb. 2003.
[8] Y. Sagduyu and A. Ephremides, “On joint MAC and network coding in wireless ad hoc networks,” IEEE Transactions on Information Theory, vol. 53, no. 10, pp. 3697 – 3713, Oct. 2007.
[9] X. Wang, W. Song, C. Yuen, and J. Li, “Exchanging Third-Party Information with Minimum Transmission Cost,” IEEE Globecom, 2012.
[10] X. Wang and C. Yuen, “Partial Third-Party Information Exchange with Network Coding,” IEEE Communications Letters, Apr 2013, pp. 757 – 760.
[11] R. Ahlswede, N. Cai, S. Li, and R. Yeung, “Network information flow,” IEEE Transactions on Information Theory, vol. 46, pp. 1204 – 1216, 2000.
[12] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: practical wireless network coding,” IEEE/ACM Transactions on Networking, vol. 16, no. 3, pp. 497 – 510, 2008.
[13] C. Fragouli, D. Katabi, A. Markopoulo, M. Medard, and H. Rahul, “Wireless network coding: opportunities & challenges,” in Proceedings of the IEEE Military Communications Conference, Oct. 2007, pp. 1 – 8.
[14] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, “Wireless broadcast using network coding,” IEEE Transactions on Vehicular Technology, vol. 58, no. 2, pp. 914 – 925, 2009.
[15] S. E. Rouayheb, M. Chudhray, and A. Sprintson, “On the minimum number of transmissions in single-hop wireless coding networks,” in IEEE Information Theory Workshop, ITW ’07, 2007, pp. 120 – 125.
[16] W. Song, X. Wang, C. Yuen, and J. Li, “Error Correction for Cooperative Data Exchange Corresponding,” IEEE Comms Letters, Nov 2012, pp. 1856 – 1859.
[17] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, “Index coding with side information,” in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2006, pp. 197C- 20.
[18] Z. Bar-Yossef, Y. Birk, T. S. Jayram and T. Kol, “Index coding with side information,” IEEE Transactions on Information Theory, Vol. 57, No. 3, March 2011.
[19] N. Alon, A. Hassidim, E. Lubetzky, U. Stav, and A. Weinstein, “Broadcasting with side information,” in Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2008, pp. 823-C832.
[20] Y. Kim and G. D. Veciana, “Is rate adaptation beneficial for inter-session network coding?” IEEE Journal on Selected Areas in Communications, vol. 27, pp. 635 – 646, Jun. 2009.
[21] K. Chi, X. Jiang, and S. Horiguchi, “Joint design of network coding and transmission rate selection for multihop wireless networks,” IEEE Transactions on Vehicular Technology, vol. 59, pp. 2435 – 2444, 2010.
[22] T. Kim, S. Vural, I. Broustis, D. Sylvestis, S. Krishnamurthy, and T. La Porta, “A framework for joint network coding and transmission rate control in wireless networks,” in Proceedings of 2010 IEEE INFOCOM, Mar. 2010, pp. 1 – 9.
[23] B. Ni, N. Santhapuri, C. Gray, and S. Nelakuditi, “Selection of bit-rate for wireless network coding,” in Proceedings of 5th IEEE SECON, 2008.
[24] D. Traskov, M. Medard, P. Sadeghi, and R. Koetter, “Joint scheduling and instantaneously decodable network coding,” in Proceedings of IEEE Globecom Workshops, pp. 1–6, 2009.
We prove it by induction.

When $k = 1$, obviously the above lemma is true. That is, $Q_1^* = Q_1^*$, and thus $S_1^* = S_1^* = \{Q_1^*\}$. For the case when $k = 2$, from the algorithms, we know that $Q_2^*$ maximizes the ratio $\frac{w_i(Q_2^*)}{c_i(Q_2^*)}$, among the rest cliques in $Q - Q_1^*$, and $Q - Q_2^*$ respectively. Since $Q_1^* = Q_1^*$, $Q_2^*$ and $Q_2^*$ are the same.

Without loss of generality, we assume that the above lemma is true for each of the first $k$ iterations, where $1 \leq k < \min\{l_q^*, l_q^*\}$. Let $U'$ and $U^*$ be the collections of unconsidered cliques left after the $k$-th iteration with the algorithm in [34] and Algorithm 5 respectively. We then consider the $(k+1)$-th iteration.

According to the definition, $Q_{k+1}^*$ and $Q_{k+1}^*$ maximize the ratio $\frac{w_i(Q_{k+1}^*)}{c_i(Q_{k+1}^*)}$ among the left cliques in $U'$ and $U^*$ respectively. Firstly, we can obtain that $U^* \subseteq U'$. This is because, at each of the first $k$ iterations, our Algorithm 5 may delete two cliques from $U^*$: one is the current considered clique, and the other is the selected clique in the current iteration, while the algorithm of [34] only deletes the current considered clique. According to the above assumption, the cliques considered in each of the first $k$ iterations with these two algorithms are the same. In addition, for each clique $Q_h$ in $U' - U^*$, the weight of the elements covered by $Q_h$ but not covered by any clique in $Q_k^*$ must be 0, which thus does not need to be considered. In other words, the clique $Q_{k+1}^*$ that maximizes the ratio of weight $w_i(Q_{k+1}^*)$ to cost among cliques in $U^*$ should be the same as the clique $Q_{k+1}^*$ that maximizes the ratio of weight $w_i(Q_{k+1}^*)$ to cost among cliques in $U_*$. Thus, the above lemma holds for the $(k+1)$-th iteration.

To summarize, we can obtain that at each $k$-th iteration, where $1 \leq k < \min\{l_q^*, l_q^*\}$, the cliques considered by the algorithm in [34] and Algorithm 5 are the same, i.e., $Q_{k+1}^* = Q_{k+1}^*$.

APPENDIX B

PROOF OF LEMMA 3

Note that according to Lemma 3 at the $k$-th iteration, we have $Q_{k+1}^* = Q_{k+1}^*$. We then prove the above lemma by considering the following two cases.

Case 1: If algorithm in [34] selects clique $Q_{k+1}^*$, there are two cases for Algorithm 5.

If there is no common element between clique $Q_{k+1}^*$ and the cliques in $Q_{k-1}^*$, $Q_{k+1}^*$ must be also selected. Otherwise, there must exist another clique, which includes the elements covered by clique $Q_{k+1}^*$ but not covered by any clique in $Q_{k-1}^*$. This is because if there does not exist such a clique, the weight $w_i(Q_{k+1}^*)$ must be 0, which contradicts that the clique $Q_{k+1}^*$ maximizes the ratio of the weight $w_i(Q_{k+1}^*)$ to cost. Thus, Algorithm 5 must also select a clique.

Case 2: If Algorithm in [34] cannot add in clique $Q_{k+1}^*$, Algorithm 5 will also discard the clique and select nothing. This is because, the only reason that the Algorithm in [34] cannot select the current clique is that adding the current clique exceeds the delay constraint $T$, which also needs to be satisfied by our Algorithm 5. In this case, both algorithm cannot add in new cliques.

To sum up, we proved Lemma 4.

APPENDIX C

PROOF OF LEMMA 5

We prove it by induction.

Firstly, when $k = 1$, we can easily obtain that the cliques in $Q_1^*$ and $Q_1^*$ cover the same set of elements, i.e., $S_1^* = S_1^*$. Without loss of generality, we assume that after the first $k-1$ iterations, the cliques in $Q_{k-1}^*$ and $Q_{k-1}^*$ cover the same set of the elements. That is, $S_{k-1}^* = S_{k-1}^*$. We then consider the case of the $k$-th iteration.

As in Lemma 4 at the $k$-th iteration, if algorithm in [34] selects the clique $Q_{k+1}^*$, there are two cases for Algorithm 5: 1) select the clique $Q_{k+1}^*$, or 2) add the clique $Q_h$ that includes the elements covered by $Q_{k+1}^*$ but not covered by any clique in $Q_{k-1}^*$, i.e., $Q_h = Q_{k+1}^* \setminus S_{k-1}^*$. For the first case, according to Lemma 4 we have $Q_{k+1}^* = Q_{k+1}^*$. As $S_{k-1}^* = S_{k-1}^*$, we can obtain that

$$S_k^* = S_{k-1}^* \cup Q_{k+1}^*$$

$$= S_{k-1}^* \cup Q_{k+1}^* = S_{k-1}^*$$

(17)

For the second case, we can obtain that

$$S_k^* = S_{k-1}^* \cup Q_h$$

$$= S_{k-1}^* \cup (Q_{k+1}^* \setminus S_{k-1}^*)$$

$$= S_{k-1}^* \cup Q_{k+1}^*$$

$$= S_{k-1}^* \cup Q_{k+1}^* = S_{k-1}^*$$

(18)
Thus, in either case, \( Q_k^* \cup (Q_{k-1}^* \setminus \bigcup (Q_{h_i}^*) \bigcup \) must cover the same set of elements as \( Q_k^* \cup (Q_{k-1}^* \setminus \bigcup (Q_{h_i}^*) \), i.e., \( S'_k = S_k^* \).

If at the \( k \)-th iteration, algorithm in \([34]\) does not add in a clique, according to Lemma 4, Algorithm 5 must also not select any new clique. That is, \( Q_k^* = Q_{k-1}^* \setminus \bigcup (Q_{h_i}^*) \). According to the above assumption, we have \( S'_k = S_k^* \).

Thus, after the \( k \)-th iteration, the cliques in \( Q_k^* \) and \( Q_k^* \) cover the same set of the elements, which thus proves Lemma 5.

**APPENDIX D**

**PROOF OF LEMMA 6**

According to lemma 4 at the \( k \)-th iteration, where \( 1 \leq k \leq \min \{l'_q, l'^*_q \} \), if algorithm in \([34]\) adds in clique \( Q_k^* \), our Algorithm 5 must also select clique \( Q_k^* \) or clique \( Q_k^* = Q_k^* \setminus Q_k^* \). According to Lemma 5, we have \( Q_k^* = Q_k^* \).

Firstly, if Algorithm 5 selects \( Q_k^* \), the cost added by both algorithms must be the same. Secondly, if Algorithm 5 selects another clique \( Q_k^* \), which includes the subset of elements in \( Q_k^* \), the cost added by Algorithm 5 must be no more than that added by clique \( Q_k^* \). In other words, at the \( k \)-th iteration, the cost added by selecting a clique with algorithm in \([34]\) is no less than that by our Algorithm 5.

Thus, Algorithm 5 can add in more cliques before exceeding the constraint \( T \), i.e., \( q' \leq q^* \). In addition, according to Lemma 4, we have \( l'_q \leq l'^*_q \).

**APPENDIX E**

**PROOF OF LEMMA 7**

From Lemma 5, we can obtain that after the first \( k = \min \{l'_q, l'^*_q \} \) iterations, the collections of the selected cliques with both algorithms, e.g., \( Q_k^* \) and \( Q_k^* \), cover the same set of elements. In other words, \( W(Q_k^*) = W(Q_k^*) \).

In addition, because \( l'_q \leq l'^*_q \), after \( k = l'_q \) iterations, the Algorithm 5 still has a chance to select more \( q^* - q' \) cliques. Hence, the total weight of the elements covered by \( Q^* \) must be less than the total weight of the elements covered by \( Q^* \), i.e., \( W(Q^*) \leq W(Q^*) \).

**APPENDIX F**

**PROOF OF THEOREM 3**

Assume that the optimal solution of budgeted set coverage problem in \([34]\) is \( OPT^* \), while the optimal solution of our pairwise coding is \( OPT^* \). The only difference between these two problems is that in our problem, the selected cliques should be disjoint with each other. Note that the optimal solution to our problem is only one of the feasible solutions to the problem in \([34]\). Thus, \( OPT^* \geq OPT^* \).

With Lemma 7, we have \( W(Q^*) \geq W(Q^*) \). According to \([34]\), we can obtain

\[
W(Q^*) \geq W(Q^*) \geq (1 - \frac{1}{e}) OPT^* \geq (1 - \frac{1}{e}) OPT^* \quad (19)
\]

Thus, the total weight achieved with our algorithm is at least \( 1 - \frac{1}{e} \) of the optimal solution for pairwise coding.

Xiumin Wang received her B.S. from the Department of Computer Science, Anhui Normal University, China, in 2006, and joint Ph.D. degree from School of Computer Science and Technology of University of Science and Technology of China and City University of Hong Kong. From 2006 to 2010, she was with the Institute for Infocomm Research, Singapore, as a Senior Research Engineer. Since 2010, she has been an Assistant Professor with the Singapore University of Technology and Design, Singapore. Dr. Yuen serves as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He received the IEEE Asia Pacific Outstanding Young Researcher Award in 2012.

Chau Yuen received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2000 and 2004, respectively. In 2005, he was a Postdoctoral Fellow with Lucent Technologies Bell Labs, Murray Hill, NJ, USA. In 2008, he was a Visiting Assistant Professor with Hong Kong Polytechnic University, Kowloon, Hong Kong. From 2006 to 2010, he was with the Institute for Infocomm Research, Singapore, as a Senior Research Engineer. Since 2010, he has been an Assistant Professor with the Singapore University of Technology and Design, Singapore. Dr. Yuen serves as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He received the IEEE Asia Pacific Outstanding Young Researcher Award in 2012.

Yinlong Xu received his B.S. in Mathematics from Peking University in 1983, and MS and Ph.D in Computer Science from University of Science and Technology of China(USTC) in 1989 and 2004 respectively. He is currently a professor with the School of Computer Science and Technology at USTC. Prior to that, he served the Department of Computer Science and Technology at USTC as an assistant professor, a lecturer, and an associate professor. Currently, he is heading a group of research students in doing some networking and high performance computing research. His research interests include network coding, wireless network, combinatorial optimization, design and analysis of parallel algorithm, parallel programming tools, etc. He received the Excellent Ph.D Advisor Award of Chinese Academy of Sciences in 2006.