QoS-Guaranteed Wireless Broadcast Scheduling With Network Coding and Rate Adaptation

Hang Shen and Guangwei Bai

***Abstract*—Network coding is critical to wireless broadcast for real-time applications. Most of the existing approaches make strong assumptions either on application requirements (e.g., single constraint type or the same constraint level) or on data transmission (e.g., ideal channel, fixed transmission rate, or packet length) in order to facilitate formalization and solution. Those assumptions limit the applicability of previous approaches. This work studies quality-of-service-guaranteed broadcast scheduling over wireless networks with network coding and rate selection capabilities, focusing on reducing broadcast completion delay while maximizing the number of packet receptions that satisfy heterogeneous deadline and reliability requirements. To begin with, a multirate graph model is constructed to formulate the optimal broadcast scheduling problem, which is proved to be NP-hard. Then, an adaptive graph compression policy is proposed to reduce the computational burden significantly without sacrificing performance. Furthermore, an approximation framework is presented for each propagation. In the framework, the coding strategy and rate selection can be formulated as a complexity-adjustable clique problem. Finally, a progressive clique search algorithm is designed to make decision on each broadcast. Simulation results demonstrate that compared with typical heuristic algorithms, the proposed algorithm achieves significant performance improvement with lower complexity.**

***Index Terms*—Broadcast scheduling, wireless networks, network coding, rate adaptation, QoS-sensitive application.**

I. Introduction

With the popularity of wireless networks and mobile computing, many emerging applications have become available, including live video broadcasting, mobile virtual reality, mobile cloud gaming and location-based services, which are very appealing for users. However, these applications have strict QoS requirements [1]–[3]. Packet loss causes communication damage such as image disturbance and freezing; a high delay causes a decrease in the availability of data such as the asynchrony of continuous location-based queries.

Studies have shown that network coding can significantly improve communication performance [4]. [5] is the first attempt to perform network coding in a wireless environment. With the scheme, each node obtains packet reception status of its neighbors by means of opportunistic listening, while relays can independently encode multiple packets with local information, thereby increasing the amount of data carried in each transmission. Wireless data distribution is an important application of network coding. While packets are distributed by a server to multiple clients by broadcast/multicast, the completion delay can be significantly reduced by using network coding. At present, technical and theoretical results have been achieved on how to minimize number [6], [7] and cost [8], [9] of broadcasts, and how to improve timeliness [10]–[13] and reliability [14], [15] in broadcast process. However, most of these schemes either depend on specific application requirements (including single constraint type or the same constraint level) or make strong assumptions on data transmission (including fixed transmission rate or packet size, or ideal channels without packet loss), which decreases practicability.

In multirate wireless networks, nodes can select transmission rate dynamically, which directly affects metrics in terms of throughput, timeliness, reliability and cost [16]–[18]. A low rate may lead to a reduction in throughput [19]; other than the decrease of delivery ratio [20], while a high rate may make adjacent nodes unable to complete listening [21]. Another challenge is heterogenous deadline and reliability requirements. Due to the complex dependencies between these performance indicators, each decision may affect its subsequent decisions. If we follow the principle of maximizing coding gain, failure of packets due to QoS constraints is very likely to occur; while using smallest-deadline-first [13] strategy reduces coding gain. These constraints and challenges, in combination with time-varying link quality and high computational complexity, make scheduling QoS-constrained broadcasts a challenging issue. Finding a way to rationally carry out rate selection and coding decision has become an important issue.

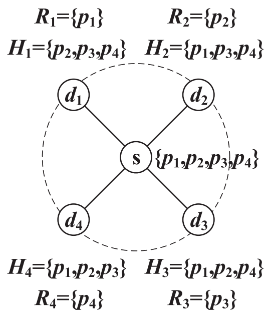
We present an example in Fig. 1 to illustrate our motivation, where server s needs to transmit packets p_1, p_2, p_3 and p_4 to four clients d_1, d_2, d_3 and d_4, respectively. Fig. 1(a) gives the set of packets owned and required at client d_i, denoted by H_i and R_i, respectively. Fig. 1(b) provides the deadline of each required packet at its destination, and the expected reception ratio (denoted by $\theta_{z,i}(t)$) for a packet transmitted over link

Manuscript received September 4, 2017; revised December 31, 2017; accepted February 8, 2018. Date of publication February 13, 2018; date of current version July 16, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61502230 and Grant 61073197, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20150960, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 15KJB520015, in part by the State Key Laboratory Program for Novel Software Technology (Nanjing University) under Grant KFKT2017B21, and in part by the Nangjing Municipal Science and Technology Plan Project under Grant 201608009. The review of this paper was coordinated by Dr. C. Yuen. *(Corresponding author: Guangwei Bai.)*

H. Shen is with the Department of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China, and also with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China (e-mail: hshen@njtech.edu.cn).

G. Bai is with the Department of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China (e-mail: bai@njtech.edu.cn).

Digital Object Identifier 10.1109/TVT.2018.2805806



(a)

packet	p_1	p_2	p_3	p_4
deadline(s)	4	9	10	15

rate(kbps) \ link	$<s, d_1>$	$<s, d_2>$	$<s, d_3>$	$<s, d_4>$
$r_1=1$	0.95	0.85	0.95	0.8
$r_2=2$	0.8	0.75	0.8	0.6
$r_3=4$	0.6	0.7	0.5	0.4
$r_4=5$	0.5	0.2	0.4	0.2
$r_5=8$	0.3	0.15	0.3	0.15
$r_6=10$	0.2	0.1	0.2	0.1

(b)

Fig. 1. Motivation illustration. (a) Packet reception status. (b) Deadlines and packet reception ratios.

$<s, d_i>$ with the z-th available transmission rate r_z at time t_0. Suppose that 1) the length of each raw packet is $L = 10$ k and 2) a redundancy scheme is incorporated in transmission to ensure the successful delivery of packets. For this scenario, a common approach according to [5], [13], [22] is to send the encoded packet $p_1 \oplus p_2 \oplus p_3 \oplus p_4$ to maximize the number of clients that can decode it. However, this is not a good choice, because selecting any one rate does not guarantee that all packets can be required in their deadlines. For example, if $r_1 = 1$ kbps is selected, the expected broadcast delay of this encoded packet is

$$\frac{L}{r_1} \cdot \max_{i \in \{1,2,3,4\}} \left\{ \frac{1}{\theta_{1,i}(t_0)} \right\} = \frac{10\ \mathrm{k}}{1\ \mathrm{k/s}} \cdot \frac{1}{0.8} = 12.5\ \mathrm{s}$$

This misses the deadlines of p_1, p_2 and p_3 since their intended destinations cannot get enough data for packet restoration in their reception deadlines. We will give a detailed quantification later. Similarly, d_1 cannot obtain p_1 in 4 s with r_2; d_2 cannot obtain p_2 in 9 s with r_4. An alternative we may choose is to first send p_1 at r_4, where d_1 will obtain p_1 in 4 s. For simplicity, we tentatively assume that the packet reception ratio remains unchanged for the 2nd broadcast. Next, if the encoded packet $p_2 \oplus p_3 \oplus p_4$ after redundancy is sent at r_3, d_2, d_3 and d_4 will obtain their wanted packets in 6.25 s. This is one of the solutions without deadline missed packets.

In this work, we study wireless broadcast scheduling with network coding and rate adaptation for QoS-sensitive application, with the objective of shortening broadcast completion delay while maximizing the number of received packets subject to heterogenous QoS constraints. The main contributions can be concluded as follows:

- A multirate graph model is proposed to determine the encoded packet and broadcast rate, and to ensure that all the output packets can be successfully received/decoded in presence of heterogenous deadline and reliability constraints. On this basis, we present an integer programming formulation of the optimal broadcast scheduling problem.
- To filter unnecessary searches in scheduling, a deadline-aware graph compression policy is proposed, followed by a problem approximation framework for each broadcast. With the framework, a progressive clique search algorithm is designed, in which the adjustment of computational complexity is realized through parameter settings.
- To demonstrate the effectiveness and practicability of our algorithm, a series of simulation experiments are conducted, where the strict degree of application requirements, the number of clients and the number of packets to be sent are taken into account. While investigating and understanding performance bottleneck, different algorithms are compared and analyzed to identify optimal solution.

The remainder of this paper is organized as follows. We briefly introduce related works in the next section. Section III describes and formulates our problem. The problem approximation framework and heuristic algorithm are given in Section IV. Finally, we present simulation results and performance analysis in Section V before concluding in Section VI.

II. Related Works

While a lot of research [23]–[26] has been devoted to scheduling broadcasts with QoS requirements over wireless networks, the proposed solutions may not be directly applied to network coded or multirate environments.

The work in [11] aims to minimize average delay for network coding based multicast, without consideration of delay guarantee. The network coding based broadcast policy in [27] can achieve asymptotic capacity with finite delay. However, it assumes that all the packet deadlines are the same. The authors in [28] address the problem of scheduling delay-constrained traffic over unreliable wireless links and propose a feasibility-optimal coding scheme. However, the impact of constraint levels is not taken into account. The broadcast scheme in [29] aims to maximize the throughput while satisfying deadlines by means of adjusting coding blocks. A reliable wireless multicast protocol is presented in [30], where the playout deadlines for packets can be met by leveraging real-time network coding. Despite the improvement in timeliness or reliability, these two works do not consider heterogenous constraints.

The authors in [31] use pairwise coding policy to schedule inter-session flows with heterogenous delay constraints and weights, focusing on maximizing the weighted sum of packets. Reference [32] presents a QoS-driven network coded multicast protocol that considers buffer overflow and delay violation constraints. A generalized encoding framework for on-demand broadcast is proposed in [13], where the optimization objective is abstracted as the clique of minimized deadline miss ratio. While it can be customized according to application requirements, the practicality is compromised due to the hypothesis of ideal channel. Reference [22] designs an encoding algorithm based on maximum weight clique for wireless broadcast with delay constraint. However, it lacks reliability support.

The above works are under the premise that the transmission rates are fixed. [20] takes into account the impact of rate selection on the expected transmission efficiency. RBAR [33] is an early rate adaptive MAC protocol for the optimum throughput with given channel conditions. [34] provides theoretical delay analysis of existing throughput-optimal coding schemes and design a rate adaptation scheme. Relatively little research has focused on QoS-constrained applications with rate adaptation. RSNC [12] is a joint rate selection and network coding scheme for maximizing the number of packets received while satisfying deadline requirements. Authors also propose a lightweight solution based on pairwise coding. Despite the assumption of ideal channel, this work provides a valuable reference for designing a delay-aware scheduling mechanism. A network coding

TABLE I
NOTATIONS AND VARIABLES

Symbols	Definition
N	set of packets to be sent
p_j	the j-th packet in N
M	set of client nodes
d_i	the i-th client in M
L	length of packet
H_i	set of packets owned by d_i
R_i	set of packets required by d_i
B	set of optional transmission rates
r_z	the z-th transmission rate in B
$\theta_{z,i}(t)$	expected reception ratio over $<s, d_i>$ with r_z at time t
t_0	broadcast start time
$a_{i,j}(t)$	the remaining deadline at time t for p_j required at d_i
$a_{i,j}$	deadline requirement of packet p_j at d_i, i.e., $a_{i,j}(t_0)$
$b_{i,j}$	reliability requirement of packet p_j at d_i
$Q_{z,h}$	the h-th optional packet-receiver relationship set for r_z
$P(Q_{z,h})$	original encoded packet formed by the packets in $Q_{z,h}$
$\mathcal{P}(Q_{z,h}, t)$	$P(Q_{z,h})$ after redundancy with r_z at time t
$\delta(Q_{z,h}, t)$	redundancy rate for $\mathcal{P}(Q_{z,h}, t)$
$T(Q_{z,h}, t)$	expected broadcast delay of $\mathcal{P}(Q_{z,h}, t)$ with r_z at time t

based reliable data dissemination approach is proposed in [35] for wireless sensor networks. The goal is to achieve energy efficiency and to minimize the completion time of data dissemination, rather than provide deadline guarantee.

After studying existing approaches in this area, we realize that there is no comprehensive study concerning providing heterogenous reliability and delay guarantees with adjustable computational complexity for network coded and multi-rate broadcast scheduling over unreliable wireless links. This drives us to purpose a practical scheme.

III. Optimal Broadcast Scheduling Problem

We first give the problem description. Then, we present a multi-rate graph model as a basis for rate selection and coding operation, followed by an analysis of scheduling decisions. Finally, we mathematically formulate the proposed problem. The main notations and variables used are listed in Table I.

A. Problem Statement

We consider a wireless broadcasting scenario. Assume that 1) there are $|N|$ packets $\{p_1, p_2, \ldots, p_{|N|}\}$ to be distributed to $|M|$ clients $\{d_1, d_2, \ldots, d_{|M|}\}$ by server s, 2) the length of each original packet is L, and 3) there are $|B|$ optional transmission rates $\{r_1, r_2, \ldots, r_{|B|}\}$. At the initial phase, each client already has a portion of packets. Denoted by H_i the set of packets owned, and $R_i = N \backslash H_i$ the set of packets to be received by client d_i. Denoted by $\theta_{z,i}(t)$ the expected reception ratio of a packet sent at r_z over link $<s, d_i>$ at time t. The packet reception ratio is detected by s by periodically broadcasting heartbeat packets and by dividing the number of packets received at one client to the number of packets sent. Rate selection is performed before each broadcast; the selected rate cannot be adjusted while a packet is being broadcasted. The reception deadline and reliability of packet p_j required at d_i is defined as $a_{i,j}$ and $b_{i,j}$, respectively. Here we evaluate the reliability by the probability that a packet is successfully delivered to its destination in a broadcast. It is assumed that s knows both of these requirements for the packet receptions at its destinations.

The XOR coding method in [5] is performed at nodes, where the packet status owned by each client can be obtained by s by using reception reports. We denote $Q_{z,h}$ to the h-th optional packet-receiver relationship set for r_z, and $P(Q_{z,h}) = p_{j_1} \oplus p_{j_2} \oplus \ldots \oplus p_{j_{|Q_{z,h}|}}$ to the original encoded packet where $(p_{j_1}, d_{i_1}), (p_{j_2}, d_{i_2}), \ldots, (p_{j_{|Q_{z,h}|}}, d_{i_{|Q_{z,h}|}}) \in Q_{z,h}$. The expression of $Q_{z,h}$ will be introduced later. Reliability for packet delivery is guaranteed by incorporating a dynamic redundancy scheme which is necessary for error resilience and can be adjusted according to the expected packet reception ratio. For a packet sent at time t, we can use the packet reception ratio recorded before t to approximate or predict $\theta_{z,i}(t)$. Denoted by $\mathcal{P}(Q_{z,h}, t)$ the $P(Q_{z,h})$ after the treatment of redundancy at time t, where the redundancy rate, denoted by $\delta(Q_{z,h}, t)$, is set to a minimum, i.e.,

$$\delta(Q_{z,h}, t) = \min\left\{\min_{(p_j, d_i) \in Q_{h,z}} \frac{\theta_{z,i}(t)}{b_{i,j}}, 1\right\} \tag{1}$$

to guarantee the reliability of p_j at d_i for each $(p_j, d_i) \in Q_{z,h}$. A smaller redundancy rate indicates more redundancy being added to a packet and better error resilience performance. The expected broadcast delay of $\mathcal{P}(Q_{z,h}, t)$ with r_z at time t is

$$T(Q_{z,h}, t) = \frac{L}{\delta(Q_{z,h}, t) \cdot r_z} \tag{2}$$

Our problem is, given H_1, R_i, $a_{i,j}$ and $b_{i,j}$ for each $p_j \in R_i$, combined with network coding and rate selection, how to shorten broadcast completion time while maximizing the number of packets received that satisfy QoS constraints.

Let $g_{i,j}$ be 1 if d_i can receive/decode p_j under given constraints, otherwise be 0, where $p_j \in R_i$. We denote $e_{i,j}$ as the delay from the start of broadcast to p_j being received/decoded by d_i if $g_{i,j} = 1$ for each $p_j \in R_i$. Accordingly, our objective can be expressed as maximizing

$$\sum_{d_i \in M} \sum_{p_j \in R_i} g_{i,j} - \alpha(\mathbb{g}, \mathbb{e}, \mathbb{a}) \tag{3}$$

where $\alpha(\mathbb{g}, \mathbb{e}, \mathbb{a})$, defined below, is a function to compress the completion of the whole process and to enhance bandwidth utilization especially when QoS constraints are less stringent.

$$\alpha(\mathbb{g}, \mathbb{e}, \mathbb{a}) = \frac{\max_{d_i \in M} \max_{p_j \in R_i} \{g_{i,j} \cdot e_{i,j}\}}{\max_{d_i \in M} \max_{p_j \in R_i} \{a_{i,j}\}} \tag{4}$$

The denominator of (4) is the upper limit of broadcast completion delay, not greater than the maximum deadline, and thus $\alpha(\mathbb{g}, \mathbb{e}, \mathbb{a}) \in [0, 1]$. With $\alpha(\mathbb{g}, \mathbb{e}, \mathbb{a})$, the achievable number of receptions will not be reduced in the compression of broadcast completion delay; in other words, if $\alpha(\mathbb{g}, \mathbb{e}, \mathbb{a})$ is removed, the purpose of (3) is merely to maximize the number of receptions and cannot further reduce broadcast completion time.

B. Multirate Graph Model

While using graph model to make network coding decisions, some previous literature [13], [22] focus on the case where the transmission rate remains fixed and thus cannot be directly tailored for our problem. The graph model in [12] considers both rate selection and deadline constraint for an ideal channel without packet loss. The graph model in [20] takes link quality at different rates into account, but with the goal of increasing expected transmission efficiency.

A multirate graph model (referred as MG) is presented in this section to formalize rate selection and coding decision for those broadcast applications with deadline and reliability constraints over unreliable wireless links.

We denote $G_z(V(t), E(t))$ to the MG for rate r_z at time t. Suppose the broadcast start time is t_0. Let $a_{i,j}(t) = a_{i,j} - (t - t_0)$ be the remaining deadline at time t for p_j required at d_i, where $t \geq t_0$. If $p_j \in R_i$ and $T(\{p_j, d_i\}, t) \leq a_{i,j}(t)$, vertex $v_{z,i,j}$ will be added to vertex set $V(G_z(t))$, given by

$$V(G_z(t)) = \{v_{z,i,j} | p_j \in R_i, T(\{p_j, d_i\}, t) \leq a_{i,j}(t)\} \quad (5)$$

According to the decoding condition, the edge set $E(G_z(t))$ consisting of $E_1(G_z(t))$ and $E_2(G_z(t))$ is expressed as

$$\begin{cases} E_1(G_z(t)) = \{(v_{z,i,j}, v_{z,i',j'} \in V(G_z(t)))| \\ \quad d_i \neq d_{i'}, p_j = p_{j'}\} \\ E_2(G_z(t)) = \{(v_{z,i,j}, v_{z,i',j'} \in V(G_z(t)))| \\ \quad d_i \neq d_{i'}, p_j \in H_{i'}, p_{j'} \in H_i\} \end{cases} \quad (6)$$

It can be inferred that: if $(v_{z,i,j}, v_{z,i',j'}) \in E_1(G_z(t)), p_j \in R_i \cap R_{i'}$; if $(v_{z,i,j}, v_{z,i',j'}) \in E_2(G_z(t))$, $p_j \in R_i \wedge p_{j'} \in R_{i'}$.

We denote $C_{z,h} = \{v_{z,i_1,j_1}, v_{z,i_2,j_2}, \ldots, v_{z,i_{|C_{z,h}|},j_{|C_{z,h}|}}\}$ to the h-th clique (complete subgraph) in $G_z(V(t), E(t))$, and $Q_{z,h} = \{(p_j, d_i) | v_{z,i,j} \in C_{z,h}\}$ to the packet-receiver relationship set involved in $C_{z,h}$. With the relationship set, the following theorem can be deduced.

Theorem 1: If s chooses r_z to broadcast $\mathcal{P}(Q_{z,h}, t)$ at time t, p_j can be received/decoded by d_i within $a_{i,j}(t)$ for each $(p_j, d_i) \in Q_{z,h}$.

Proof: There are two cases (i.e., $a_{i,j}(t) > T(Q_{z,h}, t)$ and $a_{i,j}(t) \leq T(Q_{z,h}, t)$) to be considered at time t. The former case means that $P(Q_{z,h})$ is bound to be restored within $a_{i,j}(t)$. The latter case means that if it is desired to obtain p_j within its deadline, d_i needs to receive enough data to recover $P(Q_{z,h})$ during period from t to $t + a_{i,j}(t)$. The received data volume of given d_i and r_z can be expressed as $a_{i,j}(t) \cdot r_z \cdot \theta_{z,i}(t)$, and based on this, for each $(p_j, d_i) \in Q_{z,h}$, if condition

$$a_{i,j}(t) \cdot r_z \cdot \theta_{z,i}(t) \geq L \quad (7)$$

holds, p_j can be restored by d_i without missing its deadline by means of $P(Q_{z,h}) \oplus P(Q_{z,h} \backslash \{p_j\})$. By transforming (5) and (2), we have

$$\begin{cases} a_{i,j}(t) \cdot r_z \cdot \theta_{z,i}(t) \geq T(\{p_j, d_i\}, t) \cdot r_z \cdot \theta_{z,i}(t) \\ T(\{p_j, d_i\}, t) \cdot r_z \cdot \theta_{z,i}(t) = L \cdot \frac{\theta_{z,i}(t)}{\delta(\{p_j, d_i\}, t)} = L \end{cases} \quad (8)$$

According to the transitivity in (8), condition (7) holds, and therefore the conclusion of Theorem 1 holds. ■

C. Scheduling Decision Analysis

While Theorem 1 guarantees that all the packets exported from an MG can be received/decoded within QoS constraints, the order in which encoded packets are transmitted largely affects the number of packets received due to queuing delay. We next illustrate this by means of clique partitioning. The MG for Fig. 1 is shown in Fig. 2, from which a clique with four vertexes does not exist, suggesting that if all the packets are selected

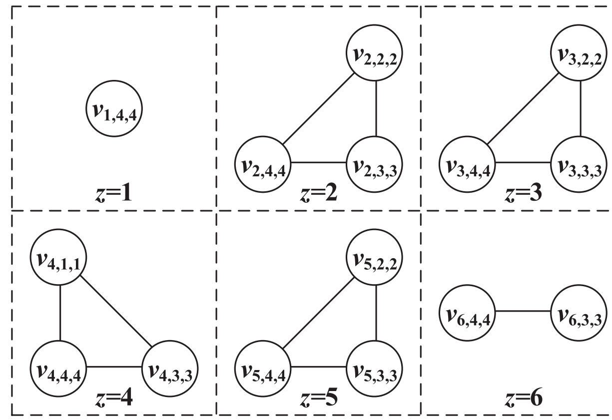


Fig. 2. The MG for Fig. 1 for the first broadcast.

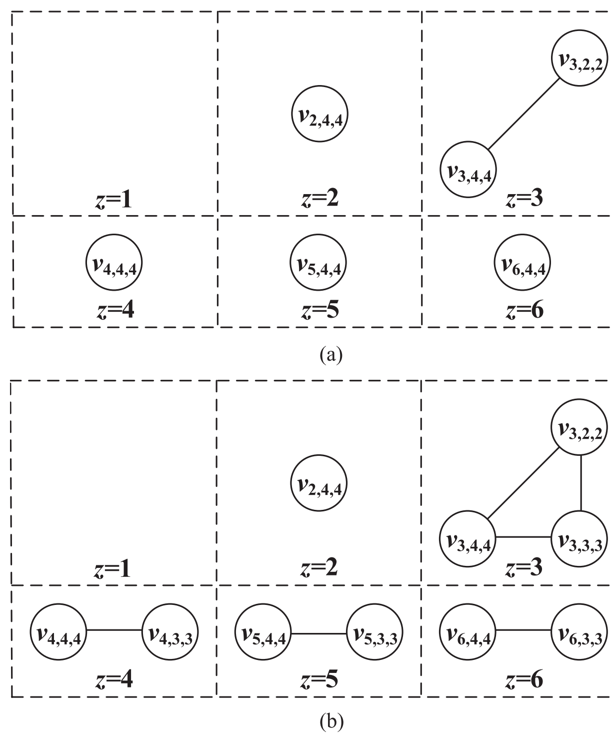


Fig. 3. The MG for Fig. 1 for the second broadcast. (a) The MG after sending $\mathcal{P}(\{p_1, p_3\}, t_0)$ at r_4 in the first broadcast. (b) The MG after sending $\mathcal{P}(\{p_1\}, t_0)$ at r_4 in the first broadcast.

to participate in encoding, choosing any transmission rate to broadcast will result in some deadline-missed packets. Other possible decisions are analyzed further below:

1) If $\{v_{3,2,2}, v_{3,3,3}, v_{3,4,4}\}$ is chosen ($\mathcal{P}(\{p_2, p_3, p_4\}, t_0)$ is sent at r_3) in the 1st broadcast (at time t_0), p_2, p_3 and p_4 can be acquired during their deadlines. According to (2), this broadcast costs 6.25 s, which exceeds $a_{1,1}$ of 4 s, resulting in the failure of p_1 before it is sent.
2) If $\{v_{4,1,1}, v_{4,3,3}, v_{4,4,4}\}$ is chosen ($\mathcal{P}(\{p_1, p_3, p_4\}, t_0)$ is sent at r_4) in the 1st broadcast, p_1, p_3, and p_4 can be received/decoded. However, the broadcast delay of 10 s exceeds $a_{2,2}$ of 9 s, resulting in the failure of p_2.

3) If $\{v_{4,1,1}, v_{4,3,3}\}$ is chosen ($\mathcal{P}(\{p_1, p_3\}, t_0)$ is sent at r_4) in the 1st broadcast, p_1 and p_3 can be acquired in 5 s. Then, packet deadlines are updated as $a_{2,2} = 9 - 5 = 4$ s and $a_{4,4} = 15 - 4 = 11$ s. If packet reception ratio does not change, the MG will updated as shown in Fig. 3(a). If $\{v_{3,2,2}, v_{3,4,4}\}$) is chosen ($\mathcal{P}(\{p_2, p_4\}, t_0 + 5)$ is sent at r_3) in the 2nd broadcast, p_2 and p_4 can be acquired. The broadcast takes a total of 11.25 s.
4) If $\{v_{4,1,1}\}$ is chosen ($\mathcal{P}(\{p_1\}, t_0)$ is sent at r_4) in the 1st broadcast, p_1 can be acquired in 4 s. Then, we have $a_{2,2} = 9 - 4 = 5$ s, $a_{3,3} = 10 - 4 = 6$ s and $a_{4,4} = 15 - 4 = 11$ s. If packet reception ratio remains unchanged, the MG will change to the form shown in Fig. 3(b). If $\{v_{3,2,2}, v_{3,3,3}, v_{3,4,4}\}$ is chosen ($\mathcal{P}(\{p_2, p_3, p_4\}, t_0 + 4)$ is sent at r_3) in the 2nd broadcast, p_2, p_3 and p_4 can be acquired. The two broadcasts take a total of 10.25 s, less than 11.25 s for the previous one. Obviously, this is the best solution found so far.

The above analysis shows that finding an optimal scheduling policy is a complex issue. While choosing a maximum clique (MC) can increase throughput per broadcast, the delay for a broadcast will increase due to more redundancy being added to the encoded packet; while giving preference to packets with shorter deadline is helpful for the reduction of the number of failed packets locally, this may degrade overall broadcast efficiency. In short, it is very difficult to attempt to achieve an overall performance boost by relying on a single indicator.

D. Problem Formulation

In this section, the optimal broadcast scheduling problem will be mathematically formalized, with an essence of finding a series of cliques, of scheduling the broadcasts of encoded packets represented by these cliques. During this process, the MG can be updated with the real-time deadlines and packet reception ratio at the end of each broadcast.

We define random variable

$$x_{z,i,j,h} = \begin{cases} 1, & v_{z,i,j} \in C_{z,h} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

The optimal solution corresponding to (3) is formulated as the following integer linear program.

Optimal Broadcast Scheduling (OBS) problem

$$\underset{\{Q_{z,h}\}}{\text{Maximize}} : \sum_{d_i \in M} \sum_{p_j \in R_i} g_{i,j} - \varphi(\mathbb{x}, \mathbb{Q}, \mathbb{t}, \mathbb{a}) \tag{10}$$

Subject to:

$$\sum_{z=1}^{|B|} \sum_{h=1}^{|V(G_z(t_0))|} x_{z,i,j,h} = 1, \forall v_{z,i,j} \in V(G_z(t_0)) \tag{11}$$

$$x_{z,i,j,h} + x_{z,i',j',h'} = 1, \forall z \in [1, |B|],$$
$$\forall h \in [1, |V(G_z(t_0))|], \forall (v_{z,i,j}, v_{z',i',j'}) \notin E(G_z(t_0)) \tag{12}$$

$$\sum_{z=1}^{|B|} \sum_{h=1}^{|V(G_z(t_0))|} \left(x_{z,i,j,h} \cdot \sum_{u=1}^{z} \sum_{w=1}^{h} T(Q_{u,w}, t_{u,w}) \right)$$
$$- \varsigma \cdot g_{i,j} \leq a_{i,j}, \forall v_{z,i,j} \tag{13}$$

$$\sum_{z=1}^{|B|} \sum_{h=1}^{|V(G_z(t_0))|} \left(x_{z,i,j,h} \cdot \sum_{u=1}^{z} \sum_{w=1}^{h} T(Q_{u,w}, t_{u,w}) \right)$$
$$+ \varsigma \cdot (1 - g_{i,j}) \geq a_{i,j}, \forall v_{z,i,j} \tag{14}$$

$$T(Q_{u,w}, t_{u,w})$$
$$= \max_{v_{z,i,j} \in V(G_u(t_0))} (x_{u,i,j,w} \cdot T(\{p_j, d_i\}_u, t_{u,w})), \tag{15}$$
$$1 \leq u \leq |\mathrm{B}|, 1 \leq w \leq |V(G_u(t_0))|$$

$$t_{u,w} = t_0 + \sum_{n=1}^{u} \sum_{m=1}^{w} T(Q_{n,m}, t_{n,m}) - T(Q_{u,w}, t_{u,w}) \tag{16}$$

$$g_{i,j} \in \{0, 1\}, \forall d_i \in M, p_j \in N \tag{17}$$

The essence of (10) is to compress total time spent on the broadcast process as much as possible without reducing the number of packets received, where the left part indicates the total number of packets within QoS constraints and the right half is a function defined as

$$\varphi(\mathbb{x}, \mathbb{Q}, \mathbb{t}, \mathbb{a})$$
$$= \frac{\sum_{z=1}^{|B|} \sum_{h=1}^{|V(G_z(t_0))|} \left(x_{z,i,j,h} \cdot \sum_{u=1}^{z} \sum_{w=1}^{h} T(Q_{u,w}, t_{u,w}) \right)}{\max_{d_i \in M} \max_{p_j \in R_i} \{a_{i,j}\}} \tag{18}$$

The numerator of (18) represents the duration of broadcast process. Similar to (4), $\varphi(\mathbb{x}, \mathbb{Q}, \mathbb{t}, \mathbb{a}) \in [0, 1]$. Constraint (11) ensures that each vertex only belongs to one clique. Constraint (12) means that a pair of vertices $v_{z,i,j}$ and $v_{z,i',j'}$ cannot belong to the same clique if there is no edge between them. Constraint (12) means that an edge exists between any pair of vertices belonging to the same clique. The large enough constant ς in constraints (13) and (14) ensures that $g_{i,j}$ is

$$\begin{cases} 1, \sum_{z=1}^{|B|} \sum_{h=1}^{|V(G_z(t_0))|} \left(x_{z,i,j,h} \cdot \sum_{u=1}^{z} \sum_{w=1}^{h} T(Q_{u,w}, t_{u,w}) \right) \geq a_{i,j} \\ 0, \sum_{z=1}^{|B|} \sum_{h=1}^{|V(G_z(t_0))|} \left(x_{z,i,j,h} \cdot \sum_{u=1}^{z} \sum_{w=1}^{h} T(Q_{u,w}, t_{u,w}) \right) \leq a_{i,j} \end{cases}$$

With the rule, the deadline for each packet in the queue can be updated after each broadcast while counting the number of successful receptions. Constraint (15) gives the delay of broadcasting $\mathcal{P}(Q_{u,w}, t_{u,w})$, which is equal to the broadcast delay with the minimum of the reception ratio among the links from s to all intended destinations. Constraint (16) indicates the start time of broadcasting $\mathcal{P}(Q_{u,w}, t_{u,w})$, which is calculated by subtracting the expected delay of broadcasting $\mathcal{P}(Q_{u,w}, t_{u,w})$ (i.e., $T(Q_{u,w}, t_{u,w})$) from the end time of broadcasting $\mathcal{P}(Q_{u,w}, t_{u,w})$ (i.e., $t_0 + \sum_{n=1}^{u} \sum_{m=1}^{w} T(Q_{n,m}, t_{n,m})$).

In the above model, each broadcast (along with a different start time) experiences different packet reception ratios; at the same time, different broadcast start times (see t_0 in (16)) mean that the start of each subsequent broadcast will be different, resulting in different local and overall performance.

Theorem 2: The OBS problem is NP-hard.

Proof: Consider a special case of the OBS problem: 1) $r_1 =$ 1 kbps is the only transmission rate available; 2) there is no packet loss; 3) for each $p_j \in R_i$, let $a_{i,j}$ be 10 and $b_{i,j}$ be 1, i.e., s can only broadcast once with fixed delay. This case is equivalent to finding an MC in a graph; meanwhile, the packets covered by the vertices outside the MC range will miss their deadlines. The NP-complete nature of this case is well known. Thus, this theorem must be true. ■

Theoretically, the optimal solution of the OBS problem could be obtained by solving the given integer linear program. However, the high computational complexity associated with the solution becomes a major limitation on performance especially when the size of an MG is large. Another hard-to-avoid problem is the time-varying quality of links. In this case, the optimal solution cannot be achieved unless the packet reception ratio for each transmission link in the future can be accurately predicted at time t_0. As only the packet reception ratio for the past period is available, a natural approach is to simplify the OBS problem for each broadcast and to design a heuristic algorithm for a suboptimal solution.

IV. Complexity-Adjustable Scheduling Algorithm

We focus on problem approximation and algorithm design in this section. We begin with an adaptive graph compression policy to improve search efficiency, building upon which the OBS problem is approximated as a complexity adjustable clique search problem for each propagation. In addressing this problem, a progressive clique search algorithm is designed to make decisions on network coding and rate selection. Finally, the implementation details of the entire broadcast process and its computational complexity will be introduced.

A. Graph Compression With Probabilistic Guarantee

In terms of the optimal solution, the probability for a packet with a long deadline to be "prior scheduled" is relatively low; in fact, "prior scheduling" takes place only when coding gains are significantly greater than the loss brought by the failure of packets. This drives us to propose an adaptive graph compression policy to filter the vertexes with longer deadlines in an MG, thereby jointly reducing unnecessary calculations and maintaining stable coding gains.

Denoted by $G_z^*(V(t),E(t))$ the probability-based MG (referred to PG in the subsequent part), which is formed after filtering the vertices with relatively long deadline by setting a threshold. The vertex set of $G_z^*(V(t),E(t))$ is defined as

$$V(G_z^*(t)) = \begin{cases} \{v_{k,i,j} \in V(G_z(t))| \\ \quad a_{i,j}(t) \le a_z^*(t)\}, \ |V(G_z(t))| > \xi \\ V(G_z(t)), \text{otherwise} \end{cases} \tag{19}$$

Equation (19) contains $a_z^*(t)$ to compress $G_z(V(t),E(t))$ as a threshold. In addition, to speed up data distribution especially at the final stage, only when $|V(G_z(t))| > \xi$ is the PG be enabled, where ξ can be adjusted to maintain a relatively large number of remaining vertices (tasks).

The essence of setting the threshold is to provide a probabilistic guarantee, in which the probability of $a_{i,j}(t) \le a_z^*(t)$ will not fall below γ, expressed by

$$P(a_{i,j}(t) \le a_z^*(t)) \ge \gamma \tag{20}$$

It can be changed to

$$P(a_{i,j}(t) > a_z^*(t)) \le 1-\gamma \tag{21}$$

Given a random variable X with mean μ and variance σ^2, according to one-sided Chebyshevs inequality, it satisfies the following inequality

$$P(X-\mu \ge l) \le \frac{\sigma^2}{\sigma^2+l^2} \tag{22}$$

We can obtain the average deadline $\overline{a_{i,j}(t)}$ and the variance of deadline $(\Delta a_{i,j}(t))^2$. By applying the one-sided Chebyshevs inequality on (21), we have

$$P(a_{i,j}(t) > a_z^*(t)) \le \frac{(\Delta a_{i,j}(t))^2}{(\Delta a_{i,j}(t))^2 + \left(a_z^*(t) - \overline{a_{i,j}(t)}\right)^2} \tag{23}$$

and

$$a_z^*(t) - \overline{a_{i,j}(t)} > 0 \tag{24}$$

According to transitive property of inequations (21) and (23), if condition

$$\frac{(\Delta a_{i,j}(t))^2}{(\Delta a_{i,j}(t))^2 + \left(a_z^*(t) - \overline{a_{i,j}(t)}\right)^2} \le 1-\gamma \tag{25}$$

holds, the probabilistic deadline guarantee (defined in (21)) could be satisfied. Inequation (25) can also be expressed as

$$a_z^*(t) \ge \overline{a_{i,j}(t)} + \Delta a_{i,j}(t) \cdot \sqrt{\frac{\gamma}{1-\gamma}} \tag{26}$$

Accordingly, the threshold can be set to

$$a_z^*(t) = \overline{a_{i,j}(t)} + \Delta a_{i,j}(t) \cdot \sqrt{\frac{\gamma}{1-\gamma}} \tag{27}$$

subject to constraint (25). During broadcast scheduling, along with vertex updates, the $a_z^*(t)$ can be dynamically adjusted with changes in mean and variance.

The value of γ determines the degree to which an MG is compressed. Specifically, if $\gamma = 0.5$, the MG is at its maximum compression; if $\gamma \to 1$, we have $G_z^*(V(t),E(t)) = G_z(V(t),E(t))$. Many unnecessary searches can be filtered out by reasonably adjusting γ. The effectiveness of this policy will be verified through simulation as described later.

B. Problem Approximation for Each Propagation

Based on the PG, a complexity-adjustable approximation framework for the OBS problem is constructed in this section.

We define random variable

$$y_{z,i,j} = \begin{cases} 1, & v_{z,i,j} \in C_z^*(t) \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

where $C_z^*(t)$ represents a clique belonging to $G_z^*(V(t),E(t))$. Let $Q_z^*(t) = \{(p_j,d_i)|v_{z,i,j} \in C_z^*(t)\}$ denote the packet-receiver relationship set from $C_z^*(t)$. The OBS problem can be approximated at each broadcast by the following LRCD problem over the rate r_z and the $Q_z^*(t)$.

Local Rate and Coding Decision (LRCD) problem

$$\underset{Q_z^*(t)}{\text{Maximize}} : \sum_{1\le z\le |\mathrm{B}|} \sum_{v_{z,i,j} \in V(G_z^*(t))} y_{z,i,j} - \phi(\mathbb{y},\mathbb{p},\mathbb{d},\mathbb{t},\mathbb{a}) \tag{29}$$

Subject to:

$$\sum_{1\leq z\leq |\mathrm{B}|}\sum_{v_{z,i,j}\in V(G_z^*(t))} y_{z,i,j} \leq k \tag{30}$$

$$y_{z,i,j} + y_{z,i',j'} \leq 1, \forall z \in [1, |B|],$$

$$\forall (v_{z,i,j}, v_{z,i',j'}) \in \overline{E(G_z^*(t))} \tag{31}$$

$$y_{z,i,j} \in \{0,1\}, \forall v_{z,i,j} \in V(G_z^*(t)) \tag{32}$$

The maximization term (29) reflects broadcast efficiency, where function $\phi(\mathtt{y},\mathtt{p},\mathtt{d},\mathtt{t},\mathtt{a})$ is defined as

$$\phi(\mathtt{y},\mathtt{p},\mathtt{d},\mathtt{t},\mathtt{a}) = \frac{\max_{1\leq z\leq |\mathrm{B}|}\max_{v_{z,i,j}\in V(G_z^*(t))}\{y_{z,i,j}\cdot T(\{p_j,d_i\},t\}}{\max\limits_{d_i\in M}\max\limits_{p_j\in R_i}\{a_{i,j}\}} \tag{33}$$

It can be inferred that $\phi(\mathtt{y},\mathtt{p},\mathtt{d},\mathtt{t},\mathtt{a}) \in [0,1]$, with which the transmission represented by the minimum expected broadcast delay among all cliques with the same size in an MG will give priority to the solution to the LRCD problem. The variable *k* included in constraint (30) determines the upper limit of $|Q_z^*(t)|$; in other words, *s* admits at most *k* packets to partake in encoding. By adjusting *k*, coding gain can be weighted and computational complexity can be adjusted. Constraints (31) and (32) belong to general constraints of clique.

Theorem 3: If there is a β-clique in $G_z^*(V(t), E(t))$ and $\beta \leq k$, then $k \geq |Q_z^*(t)| \geq \beta$.

Proof: The condition $\phi(\mathtt{y},\mathtt{p},\mathtt{d},\mathtt{t},\mathtt{a}) \in [0,1]$ guarantees that the weight of any $(\beta-1)$-clique is not greater than the weight of a β-clique. Thus, the theorem holds. ∎

Remark 1: Theorem 3 indicates: any smaller (β - 1)-clique do not need to be considered if a β-clique exists. In practice, this property can be used to reduce complexity of clique searching.

Some implied relationships between constraint (30) and problem solving can be observed as follows:

Observation 1: Without constraint (30) (or when $k \to +\infty$), the LRCD problem is translated into a special MC problem. All the MCs need to be generated, and which one to choose depends on $\phi(\mathtt{y},\mathtt{p},\mathtt{d},\mathtt{t},\mathtt{a})$. The algorithm proposed in [36], with a worst-case time complexity of $O(3^{|V(G_z^*(t))|/3})$ given $G_z^*(V(t), E(t))$, can be used to solve such problem.

Observation 2: With constraint (30), the LRCD problem can be considered as a size-constrained clique (referred as SC) problem rather than the existing *k*-clique (KC) problem or maximum weight *k*-clique (MWKC) problem (the input is an undirected graph and a number *k*; the output is a clique with *k* vertices, if one exists) [37]; especially, when $k = 2$, the LRCD problem is converted into pairwise coding with the lowest computational complexity.

Because $|C_z^*(t)|$ is unknown, the LRCD problem cannot be directly solved by applying the existing MC, KC or MWKC algorithm. It is therefore necessary to design an adaptive clique search algorithm for constraint (30).

C. Progressive Clique Search Algorithm

A progressive clique search algorithm is designed to solve the LRCD problem. The algorithm execution process is shown in Algorithm 1. Let $\Omega_{z,\beta}$ denote the set for β-clique in $G_z^*(V(t), E(t))$. At first, we need to find all the 2-cliques by enumerating all the edges (see line 5). Each found 2-clique in $G_z^*(V(t), E(t))$ is placed in $\Omega_{z,2}$ as a reference for subsequent clique searches. Specifically, the elements contained in $\Omega_{z,\beta}$ can be used to find larger (β + 1)-cliques by comparing all the pairs of β-cliques. If any pair of elements belonging to $\Omega_{z,\beta}$ has $\beta - 1$ vertices in common and the graph contains the missing edge (see line 11), we can form a (β + 1)-clique and put it in $\Omega_{z,\beta+1}$ (see line 12). Repeat the execution until β equals to k or there is no larger (β + 1)-clique (see line 14). The final β for each rate will be recorded, with the maximum being expressed as π (see line 16) which is equal to $|C_z^*(t)|$. Finally, by comparing all the π-cliques in $\Omega_{z,\pi}$ for each $z \in B$), we choose the clique with the shortest expected broadcast delay to determine an encoded packet and rate it is broadcasted.

Worth noting that, when a β-clique is found, the probability of $|C_z^*(t)| < \beta$ is excluded by Theorem 3. This ensures the execution of the progressive search, in which a clique of size less than β can be skipped.

Algorithm 1: search(G, k, t).

1 $\rho \leftarrow \infty$; $\pi \leftarrow 1$;
2 **for** $z \leftarrow 1$ **to** $|B|$ **do**
3   **for** $\tau \leftarrow 2$ **to** k **do**
4         $\Omega_{z,\tau} \leftarrow \emptyset$;
5   **foreach** $(v_{z,i,j}, v_{z,i',j'}) \in E(G_z(t))$ **do**
6         $\Omega_{z,2} \leftarrow (v_{z,i,j}, v_{z,i',j'})$;
7   **if** $\Omega_{z,2} \neq \emptyset$ **then**
8         $\beta \leftarrow 2$;
9   **while** $\beta < k$ **do**
10   **foreach** $\omega, \omega' \in \Omega_{z,\beta}$ **do**
11   **if** $|\omega \cap \omega'| == \beta - 1\&\&$ $(\omega \cup \omega')\backslash(\omega \cap \omega') \in \Omega_{z,2}$ **then**
12                     $\Omega_{z,\beta+1} \leftarrow \omega \cup \omega'$;
13   **if** $\Omega_{z,\beta+1} == \emptyset$ **then**
14   Break;
15             $\beta \leftarrow \beta + 1$;
16   **if** $\pi > \beta$ **then**
17             $\pi \leftarrow \beta$;
18 **for** $z \leftarrow 1$ **to** $|B|$ **do**
19   **foreach** $\omega \in \Omega_{z,\pi}$ **do**
20         $\varpi \leftarrow \{(p_j, d_i)|v_{z,i,j} \in \omega\}$;
21   **if** $\rho > T(\varpi, t)$ **then**
22             $\rho \leftarrow T(\varpi, t)$;
23             $r_c \leftarrow r_z$;
24             $Q_c^*(t) \leftarrow \varpi$;
25 **return** $\{r_c, Q_c^*(t)\}$;

D. Two-Level Complexity Optimization

While an encoded packet and broadcast rate for each data distribution are determined by Algorithm 1, the entire broadcast process with multiple separate broadcasts is completed on the server by Algorithm 2. The input parameters γ and k (defined in (20) and (30)) for the function $schedule(\cdot)$ co-plays the role

Algorithm 2: schedule(γ, k, t_0).

1 $t \leftarrow t_0$;
2 **while** $\bigcup_{z\in B} V(G_z(t)) \neq \emptyset$ **do**
3   **for** $z \leftarrow 1$ **to** $|B|$ **do**
4     $a_z^*(t) \leftarrow \overline{a_{i,j}(t)} + \Delta a_{i,j}(t) \cdot \sqrt{\gamma/1-\gamma}$;
5   construct $G_z^*(V(t), E(t))$ with $a_z^*(t)$;
6   $\{r_c, Q_c^*(t)\} \leftarrow$search($\bigcup_{z\in B} G_z^*(V(t), E(t))$, k, t);
7   broadcast $P'(Q_c^*(t), t)$ at r_c;
8   **foreach** $d_i \in M$ **do**
9   **foreach** $p_j \in R_i$ **do**
10       $R_i \leftarrow R_i \setminus \{p_j\}$;
11       $H_i \leftarrow H_i \setminus \{p_j\}$;
12       $a_{i,j} \leftarrow a_{i,j} - T(Q_c^*(t), t)$;
13   $t \leftarrow t_0 + T(Q_c^*(t), t)$;

of adjusting computational complexity. If the vertex set is not empty (indicating that one or more packets are waiting to be sent), a PG will be established according to the given threshold (see lines 4 and 5). Every generation of the PG is accompanied by an update of packet deadlines, where the packets that are bound to miss their deadlines are removed. With the PG, the function $search(\cdot)$ (defined in Algorithm 1) is performed to feed back the intended transmission rate (i.e., r_c) and the packet-receiver relationship set (i.e., $Q_c^*(t)$) for the generation of $P(Q_c^*(t))$. After broadcasting $\mathcal{P}(Q_c^*(t), t)$ at r_c, the vertices belong to $Q_c^*(t)$ will be removed (see lines 9 and 10) as part of the update of the set of vertices. This process will be repeated until the vertex set is empty.

E. Algorithm Complexity Analysis

The complexity for single packet propagation is determined by Algorithm 1 with the objective of finding a size-constrained clique while maximizing broadcast efficiency. Different from judging whether a graph contains a k-clique (i.e., clique decision problem which is NP-complete), finding the target clique (i.e., $C_z^*(t)$) in $G_z^*(V(t), E(t))$ is a polynomial problem which runs in $O(|V(G_z^*(t))|^{|C_z^*(t)|})$ time. Due to the need to traverse all available rates in B, the total complexity of single broadcast becomes $O(\sum_{z=1}^{|B|} |V(G_z^*(t))|^{|C_z^*(t)|})$. Because the upper limit of the number of propagations is mainly determined by $|N|$, the algorithm complexity for the whole transmission process can be expressed as $O(\sum_{z=1}^{|B|} \sum_{h=1}^{|N|} |V(G_z^*(t_{z,h}))|^{|C_z^*(t_{z,h})|})$.

V. Performance Evaluation

In this section, performance analysis and evaluation of the proposed scheme are conducted in simulation methodology. For each $d_i \in M$, the amount of packets belonging to H_i and R_i is allocated randomly, which satisfies $H_i \cap R_i = \emptyset$. By default, the server has an optional transmission rate (kbps) of 1, 2, 4, 5, 8 and 10. The packet reception ratio of the link between the server and every reception node for different transmission rates is set by referring to [33]. The length of the data packet is $L =$ 10 k. Without loss of generality, the reliability requirements of the data packet are all set to 1.

TABLE II
LRCD With Different Parameters

Classification	schedule(γ, k, t_0)		Problem abstract
LRCD-1	$\gamma \rightarrow 1$	$k \rightarrow +\infty$	MC
LRCD-2	$\gamma = 0.5$	$k \rightarrow +\infty$	MC with PG
LRCD-3	$\gamma = 0.5$	$k = 4$	SC with PG
LRCD-4	$\gamma = 0.5$	$k = 3$	SC with PG
LRCD-5	$\gamma = 0.5$	$k = 2$	SC with PG (pairwise coding)

To observe the impact of parameter settings on broadcast performance, LRCD strategies are divided into five categories by setting γ and k, the details of which are summarized in Table II. From LRCD-1 to LRCD-5, their computational complexity shows a decreasing trend. This allows us to find an optimal combination between complexity and performance.

For performance comparison, we choose two baselines, i.e., earliest-deadline-first (EDF) [13] and smallest-deadline-maximum-number-first (SMF) [22], both of which belong to heuristic coding algorithms based on maximum weight clique search, with the difference being the vertex weight whose value reflects the probability of the packet being selected. EDF emphasizes that the packets with smaller deadline should be selected as earlier as possible, while SMF highlights the need to increase the number of packets involved in encoding in addition to the features of EDF. Because neither takes into account the rate selection and unreliable links, we extend the capabilities of both in terms of rate adaptation and dynamic redundancy so as to make them comparable.

Three experiments are designed to study the effect of the strict degree of QoS constraints, the number of receivers, and number of packets to be sent on broadcast scheduling performance. In order to improve the accuracy of experimental results, all the data presented includes the average of 200 random experiments. The performance metrics to be examined are as follows: 1) received packets: the total number of packets received/decoded by clients under given QoS constraints; 2) miss ratio: the proportion of packets that miss application requirements out of the total number of packets required by clients; 3) packets per second: the number of packets received by clients per second, reflecting the bandwidth utilization of the entire data distribution process.

A. Impact of Application Requirements

Experiment I looks at the effect of application requirements on broadcast performance, the results of which is shown in Fig. 4. The number of receivers $|M|$ and the number of packets $|N|$ are fixed to 8 and 10, respectively; packet deadlines are randomly generated within the range of $[1, a]$, and its upper limit is set to a in the range of [60, 140], with a step size of 20. To further analyze the effect of the number of optional transmission rates on the performance of different algorithms, we present the results for both $|B| = 6$ and $|B| = 2$. The former is allowed to select all six rates while the latter is only allowed to choose either 1 or 2 bps.

Fig. 4(a) shows the variation of the number of packets received by each algorithm. As a increases, the QoS constraint tends to become less strict gradually, and the number of packet reception continues to increase. The advantages of LRCD-1 and

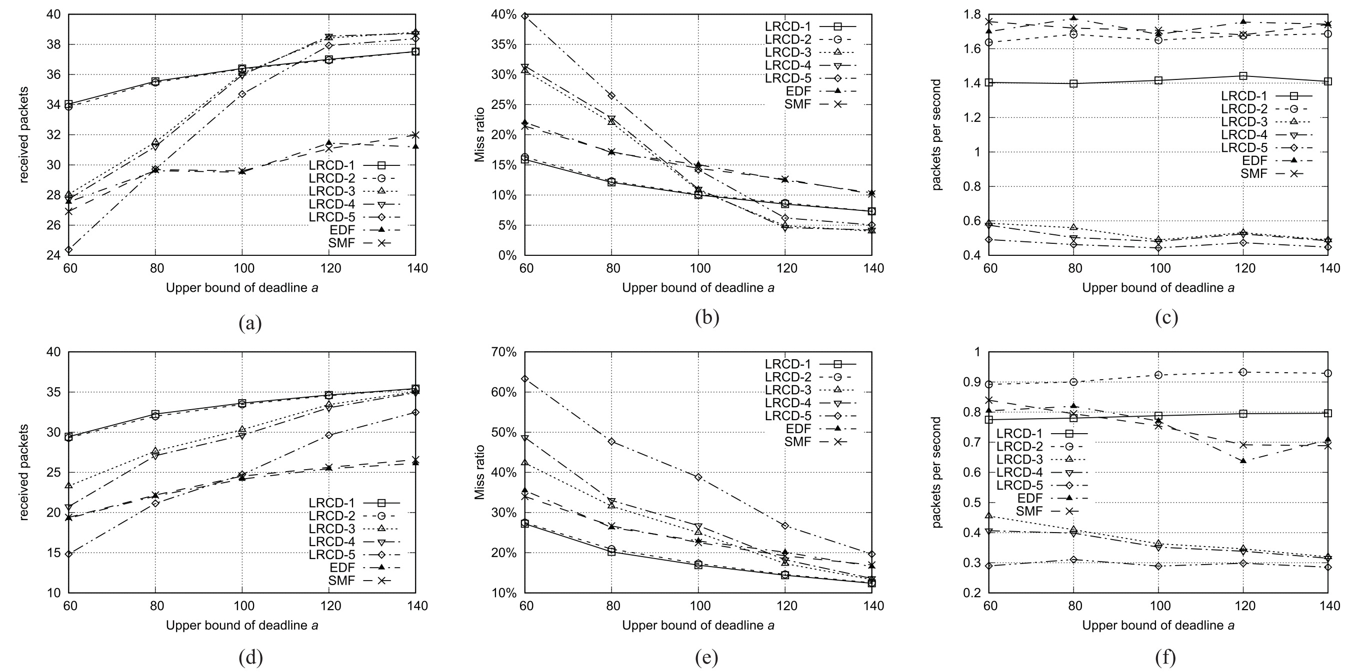


Fig. 4. Impact of application requirements (Experiment I). (a) Number of received packets ($|B| = 6$). (b) Miss ratio ($|B| = 6$). (c) Packets per second ($|B| = 6$). (d) Number of received packets ($|B| = 2$). (e) Miss ratio ($|B| = 2$). (f) Packets per second ($|B| = 2$).

LRCD-2 are more pronounced when the QoS constraint is more stringent ($a \in \{60, 80\}$). This is because when a large number of packets form a backlog, the greedy character of the MC search enables the completion of more transmission tasks at the early stage. Unlike the first two ones, LRCD-3, LRCD-4 and LRCD-5 show superiority when the QoS constraint becomes less strict ($a \in \{120, 140\}$). One important reason for this is that the broadcast delay of them is relatively short, thereby reducing its drain on other packets' deadlines. The performance gains of EDF and SMF are both at a low level. Although more packets may partake in encoding, more redundancy may be added into the encoded packet, due to the lack of consideration in link quality. This increases the risk of failure of other packets to be sent. The reason why the two algorithms is slightly higher than LRCD-5 is that they have obvious advantages in coding gain compared with pair-wise coding adopted by LRCD-5.

As can be seen in Fig. 4(b), the increase in a makes the QoS constraint less strict, and the miss ratio shows a downward trend. When the constraint is strict, the performance bottleneck is the number of broadcasts, which determines the upper limit of the number of packets received. Because higher coding gain can be achieved by LRCD-1 and LRCD-2 in each broadcast, their performance is significantly better than those of other algorithms. The comparison between the two shows that the performance is not compromised using LRCD-2 with lower computational complexity, which proves the validity of the graph compression method in Section IV-A. From another point of view, if the packets with relatively long deadlines are broadcasted during the early stage of the process, the risk of failure for short-lived packets will increase. Meanwhile, if the constraint is less strict ($a \in \{120, 140\}$), the performance bottleneck is no longer the number of broadcasts. Because the encoded packet generated by LRCD-1 and LRCD-2 is highly redundant, the expected delay for each broadcast is higher than other LRCD algorithms, which obviously squeezes the deadline of subsequent packets, thereby causing some of the packets to fail before being sent. In this regard, the three LRCD algorithms with the lowest computational complexity are found to perform better than the other strategies. The trend of EDF/SMF is close to that of LRCD-1/LRCD-2, but the latter show significant gain over the former. This is because at the same clique size, the broadcast delay of the latter is often lower than that of the former.

Fig. 4(c) gives the results on packets per second. We can see that as a increases, the performance remains relatively stable. In particular, when the QoS constraint becomes less strict, although the miss ratios of LRCD-3, LRCD-4 and LRCD-5 are lower than those of LRCD-1 and LRCD-2, the required number of broadcasts becomes greater, so their bandwidth utilization is still lower than those of LRCD-1 and LRCD-2. Further observation shows that the packets per second achieved by LRCD-2 is significantly greater than that of LRCD-1, which once again proves the practicability of our graph compression policy. One interesting phenomenon for EDF/SMF is that while the number of packets received is low, their packets per second is high. In the experiment, we found that there is a small difference in the number of broadcasts between EDF/SMF and LRCD-1/LRCD-2, but the former produces more failure data during broadcasting, resulting in a shorter broadcast duration. This, on the contrary, promotes the improvement of packets per second. Despite the consideration of the number of receivers (vertices), the trend of SMF is still closer to that of DEF due to the large weight of deadline.

It can be seen from Fig. 4(d)–(e) the overall broadcast performance degrades significantly because of the limited number of available transmission rates. Although the performance gap among algorithms becomes smaller, the polylines becomes eas-

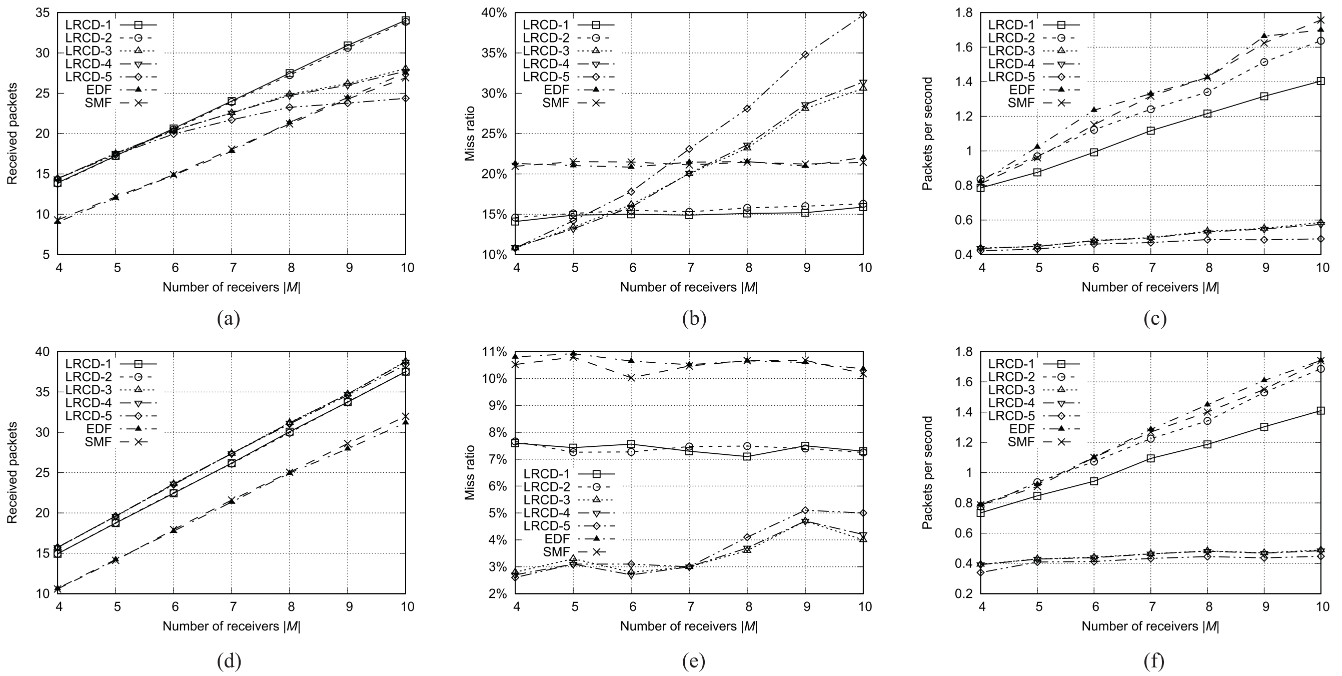


Fig. 5. Impact of number of receivers *m* (Experiment II). (a) Number of received packets ($a = 60$). (b) Miss ratio ($a = 60$). (c) Packets per second ($a = 60$). (d) Number of received packets ($a = 140$). (e) Miss ratio ($a = 140$). (f) Packets per second ($a = 140$).

ier to distinguish. Unlike Fig. 4(a), the number of receptions of LRCD-1 and LRCD-2 shown in Fig. 4(d) is maintained at a high level. From Fig. 4(e), the miss ratio of EDF/SMF is always higher than that of LRCD-1/LRCD-2; as a increases, the miss ratio gradually exceeds that of LRCD-3 and LRCD-4. In Fig. 4(f), the advantage of EDF/SMF on packets per second is significantly diminished. The main reason for this is that the reduction in the number of selectable rates reduces the potential to utilize rate selection to improve performance.

B. Impact of Number of Receivers

The purpose of the next experiment is to analyze impact on the performance exerted by the increase in $|M|$ when $|N|$ is fixed to 8. The results shown in Fig. 5 are divided into two parts according to the upper limit of deadline. In general, the increase in the number of clients is conducive to increasing coding opportunities and gain. Due to the limited clique size, algorithms like LRCD-5 are more sensitive to the increase in the number of clients; on the contrary, algorithms based on MC or MWC can take advantage of the increased number of clients to accelerate data distribution.

Fig. 5(a) shows the number of the received packets when $a = 60$. The trends of these algorithms are basically the same if the value of $|M|$ is relatively small. With the increase of $|M|$, more packets are queued for transmission, and the bottleneck of network performance is gradually presented, showing the superiority of LRCD-1. This is because the increase in $|M|$ means more coding opportunities, i.e., higher probability for LRCD-1 and LRCD-2 to achieve a high coding gain. Although the latter is less complex, its performance is better than the former. In contrast to the other three algorithms, their advantages in broadcast delay are not enough to make up for their disadvantages in coding gain. Both EDF and EMF emphasize timeliness while ignoring the delay optimization for each broadcast, thus reducing the broadcast efficiency. This is evidenced by the experimental results.

Turning to Fig. 5(d), the trends of all strategies are basically the same, which differ from Fig. 4(a). To be specific, the number of packets received by LRCD-1 and LRCD-2 are less than other three strategies. This is due to a less strict requirement that allows the server to spend more on the number of broadcasts to complete the data distribution. Nevertheless, for LRCD-1 and LRCD-2, the risk of failure of packets to be sent is increased due to longer delay required for each broadcast.

As shown in Fig. 5(b) and (e), different from Experiment I, LRCD-1 and LRCD-2 have advantages in miss ratio when $|M|$ is high and a is low, whereas the other three LRCD strategies have advantages. We can see that the polylines of LRCD-3, LRCD-4 and LRCD-5 show a rapid upward trend in Fig. 5(b). This upward trend is weakened in Fig. 5(e) but still exists. We can infer that in the face of a large number of clients, the gain from in broadcast delay such as LRCD-5 is not enough to offset their loss in coding gain.

The results on packets per second are shown in Fig. 5(c) and (f). Unlike Experiment I, the performance of EDF is optimal. In fact, the increase in the number of clients is a double-edged sword for EDF. The increase in coding gain inevitably accompanies the increase in broadcast delay, which leads to the premature termination of the broadcasting process due to the expiration of a considerable part of packets. This contributes to the improvement of packets per second.

C. Impact of Number of Packets

In Experiment III, the performance under varying number of packets $|N|$ is explored, where $|M|$ is fixed to 10. The results taking into account both $a = 60$ and $a = 140$ are shown in Fig. 6. From the perspective of an MG, an increase either $|N|$ or $|M|$

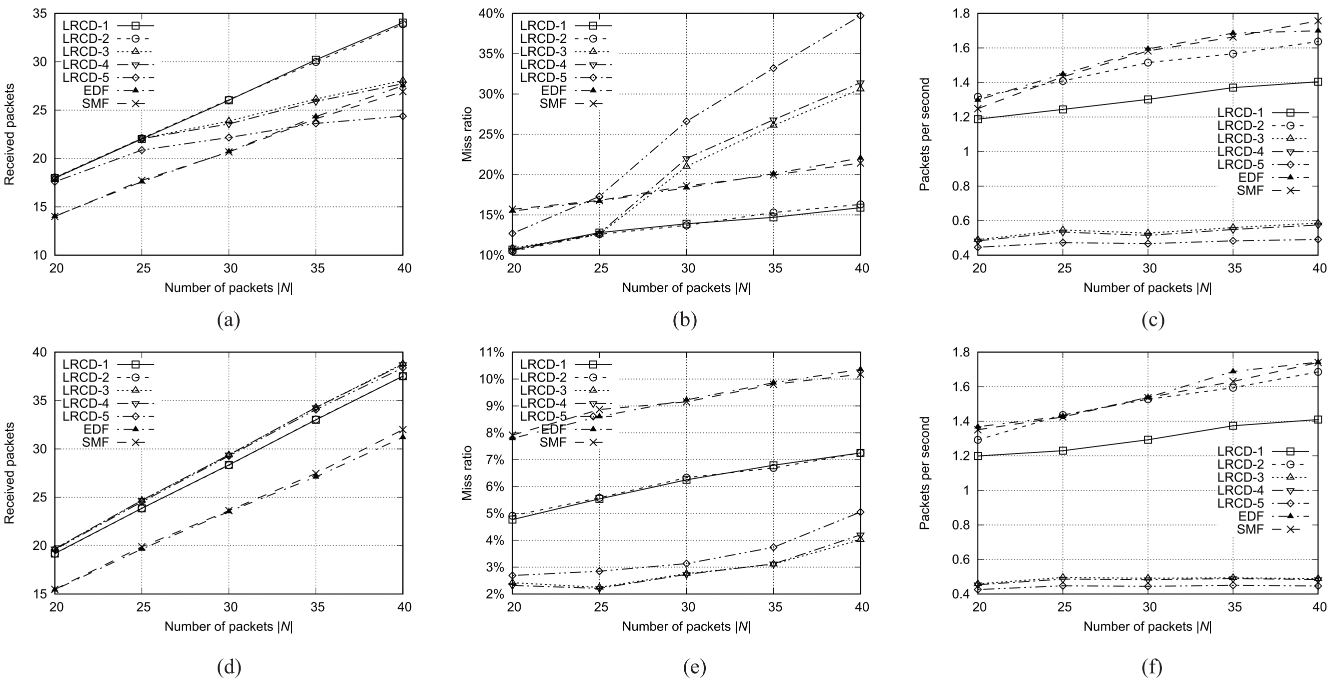


Fig. 6. Impact of number of receivers *m* (Experiment III). (a) Number of received packets ($a = 60$). (b) Miss ratio ($a = 60$). (c) Packets per second ($a = 60$). (d) Number of received packets ($a = 140$). (e) Miss ratio ($a = 140$). (f) Packets per second ($a = 140$).

can cause the number of vertices to increase. The difference is that an increase in $|M|$ causes the number of sides to increase (i.e., more opportunities for coding) while the increase in $|N|$ does not greatly affect the number of edges.

By comparing Fig. 5(a) and 6(a), it is found that with the increase of $|N|$, these polylines corresponding to the latter are more divergent. The reason is that $|M|$ is greater in this experiment (equal to the upper limit of $|M|$ in Experiment II), which will help LRCD-1 and LRCD-2 produce higher coding gains. In comparison, the coding gain of the other three LRCD strategies is limited by k. As $|N|$ increases, the difference will be further amplified. The polylines in Fig. 5(b) show a more pronounced upward trend compared to that in Fig. 6(b), as an increase in $|N|$ exacerbates the server's communication pressure without any other gains such as coding.

In terms of miss ratio, the trends in Figs. 5(c) and 6(c) are different in the early stages, because a greater $|M|$ means a greater packet base and a smaller $|N|$ will also result in performance bottlenecks. The characteristics of the polylines in Figs. 5(d) and 6(d) are not quite the same. As $|N|$ increases, the miss ratio of the former shows an upward trend while the latter does not change significantly. This is also caused by the performance bottleneck brought about by high communication pressure. The results on the change of packets per second shown in Figs. 6(c) and 6(f) are similar to those of the previous two ones, where the bandwidth utilization of EDF is the highest, next is SMF, followed by LRCD-2.

Our experimental results so far demonstrate that the superiority of the proposed LRCD scheme. In conclusion, LRCD-1 can maximize coding gain, but its overall performance is not optimal. Especially when QoS constraints are less strict, in terms of the number of packet receipts and miss ratio, LRCD-1 is inferior to LRCD-5 which has the lowest computational complexity. When QoS constraints are rather strict, LRCD-2 is more suitable; despite the limited coding gain, LRCD-5 has great performance in specific environments, especially when QoS constraints are loose or the number of clients is small. EDF/SMF cannot adapt to a wireless network with unreliable links. We can choose a suitable algorithm according to network scenarios and application requirements.

VI. Conclusion

We have proposed a complexity-adjustable wireless broadcast scheduling algorithm that combines network coding and rate adaptation. The purpose is to reduce completion time of broadcast data distribution while maximizing the number of packets that are received under QoS constraints. First, a multi-rate graph is built to model the optimal broadcast scheduling problem, and the NP-hard nature of this problem is proved. Then, in order to reduce unnecessary calculations, a deadline-aware graph compression is presented, working together with a problem approximation framework for each broadcast. Finally, a progressive clique search algorithm is designed to determine an encoded packet and rate with which it is broadcasted. Simulation results show that the proposed algorithm achieves significant improvement of broadcast efficiency with lower complexity in comparison with existing heuristic coding algorithms. We believe that our results could contribute to the design of efficient broadcast data distribution for QoS-sensitive applications over wireless networks.

Acknowledgment

The authors would like to thank the anonymous reviewers who provided excellent and thorough feedback that improved the quality of this paper.

REFERENCES

[1] A. Le, A. S. Tehrani, A. Dimakis, and A. Markopoulou, "Recovery of packet losses in wireless broadcast for real-time applications," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 676–689, Apr. 2017.

[2] X. Zheng, Z. Cai, J. Li, and H. Gao, "A study on application-aware scheduling in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 7, pp. 1787–1801, Jul. 2017.

[3] H. Li, B. Li, T. T. Tran, and D. C. Sicker, "Transmission schemes for multicasting hard deadline constrained prioritized data in wireless multimedia streaming," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 1631–1641, Mar. 2016.

[4] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, "Network coding theory: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1950–1978, Fourth Quarter 2013.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 497–510, Jun. 2008.

[6] L. Huang and C. W. Sung, "Scheduling and network coding for relay-aided wireless broadcast: Optimality and heuristic," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 674–687, Feb. 2014.

[7] M. J. Neely, A. S. Tehrani, and Z. Zhang, "Dynamic index coding for wireless broadcast networks," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7525–7540, Nov. 2013.

[8] F. Zhao, M. Médard, A. Ozdaglar, and D. Lun, "Convergence study of decentralized min-cost subgraph algorithms for multicast in coded networks," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 410–421, Jan. 2014.

[9] M. A. Raayatpanah, H. S. Fathabadi, B. H. Khalaj, and S. Khodayifar, "Minimum cost multiple multicast network coding with quantized rates," *Comput. Netw.*, vol. 57, no. 5, pp. 1113–1123, Apr. 2013.

[10] C.-C. Wang, "Delay-constrained capacity for broadcast erasure channels: A linear-coding-based study," in *Proc. IEEE Int. Symp. Inf. Theory*, 2016, pp. 2903–2907.

[11] W.-L. Yeow, A. T. Hoang, and C.-K. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *Proc. IEEE INFOCOM*, 2009, pp. 190–198.

[12] X. Wang, C. Yuen, and Y. Xu, "Coding-based data broadcasting for time-critical applications with rate adaptation," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2429–2442, Jun. 2014.

[13] C. Zhan, V. C. Lee, J. Wang, and Y. Xu, "Coding-based data broadcast scheduling in on-demand broadcast," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3774–3783, Nov. 2011.

[14] A. Tassi, I. Chatzigeorgiou, and D. E. Lucani, "Analysis and optimization of sparse random linear network coding for reliable multicast services," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 285–299, Jan. 2016.

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

[16] G. Wang, S. Zhang, K. Wu, Q. Zhang, and L. M. Ni, "TiM: Fine-grained rate adaptation in WLANs," *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 748–761, Mar. 2016.

[17] A. Fu, P. Sadeghi, and M. Médard, "Dynamic rate adaptation for improved throughput and delay in wireless network coded broadcast," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1715–1728, Dec. 2014.

[18] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proc. ACM SIGCOMM*, 2014, pp. 187–198.

[19] T.-S. Kim *et al.*, "Realizing the benefits of wireless network coding in multirate settings," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 950–962, Jun. 2013.

[20] K. Chi, X. Jiang, and S. Horiguchi, "Joint design of network coding and transmission rate selection for multihop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2435–2444, Jun. 2010.

[21] Y. Kim and G. De Veciana, "Is rate adaptation beneficial for inter-session network coding?" *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 635–646, Jun. 2009.

[22] C. Zhan and F. Xiao, "Coding based wireless broadcast scheduling in real time applications," *J. Netw. Comput. Appl.*, vol. 64, pp. 194–203, Apr. 2016.

[23] X. Kang, I.-H. Hou, and L. Ying, "On the capacity requirement of largest-deficit-first for scheduling real-time traffic in wireless networks," in *Proc. ACM MobiHoc*, 2015, pp. 217–226.

[24] K. S. Kim, C.-P. Li, and E. Modiano, "Scheduling multicast traffic with deadlines in wireless networks," in *Proc. IEEE INFOCOM*, 2014, pp. 2193–2201.

[25] G. Pei, S. Parthasarathy, A. Srinivasan, and A. K. S. Vullikanti, "Approximation algorithms for throughput maximization in wireless networks with delay constraints," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1988–2000, Dec. 2013.

[26] I.-H. Hou and R. Singh, "Scheduling of access points for multiple live video streams," in *Proc. ACM MobiHoc*, 2013, pp. 267–270.

[27] X. Li, C.-C. Wang, and X. Lin, "Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints," in *Proc. IEEE INFOCOM*, 2010, pp. 1–5.

[28] 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.

[29] L. Yang, Y. E. Sagduyu, J. Zhang, and J. H. Li, "Deadline-aware scheduling with adaptive network coding for real-time traffic," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1430–1443, Oct. 2015.

[30] D. Ferreira, R. A. Costa, and J. Barros, "Real-time network coding for live streaming in hyper-dense WiFi spaces," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 773–781, Apr. 2014.

[31] Y. Mao, C. Dong, H. Dai, X. Zhu, and G. Chen, "Optimal scheduling with pairwise coding under heterogeneous delay constraints," in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 385–390.

[32] W. Pu, C. Luo, F. Wu, and C. W. Chen, "QoS-driven network coded wireless multicast," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5662–5670, Nov. 2009.

[33] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. ACM MobiCom*, 2001, pp. 236–251.

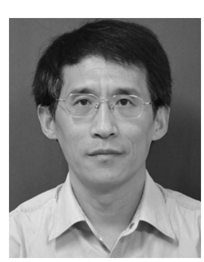
[34] A. Fu, P. Sadeghi, and M. Médard, "Dynamic rate adaptation for improved throughput and delay in wireless network coded broadcast," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1715–1728, Dec. 2014.

[35] X. Wang, J. Wang, and Y. Xu, "Data dissemination in wireless sensor networks with network coding," *EURASIP J. Wireless Commun. Netw.*, vol. 2010, pp. 1–14, Jun. 2010.

[36] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theor. Comput. Sci.*, vol. 363, no. 1, pp. 28–42, Oct. 2006.

[37] V. Vassilevska, "Efficient algorithms for clique problems," *Inf. Process. Lett.*, vol. 109, no. 4, pp. 254–257, Jan. 2009.

Hang Shen received the Ph.D. (Hons.) degree in computer science from the Nanjing University of Science and Technology, Nanjing, China. Since 2015, he has been an Assistant Professor with the Department of Computer Science and Technology, Nanjing Tech University, Nanjing. His research interests include network coding, wireless networks, edge computing, and data privacy. He is a member of the Association for Computing Machinery.



Guangwei Bai received the B.Eng. and M.Eng. degrees in computer engineering from Xi'an Jiaotong University, Xi'an, China, in 1983 and 1986, respectively, and the Ph.D. degree in computer science from the University of Hamburg, Hamburg, Germany, in 1999. From 1999 to 2001, he was a Research Scientist at the German National Research Center for Information Technology, Germany. In 2001, he joined the University of Calgary, Canada, as a Research Associate. Since 2005, he has been a Professor of computer science at Nanjing Tech University, Nanjing, China. In 2011, he was a Visiting Professor with the University of Waterloo, Canada. His research interests include wireless networks, multimedia, edge computing, and location-based services. He is a member of the Association for Computing Machinery.