Coding Based Broadcast for Layered Video Streaming in Wireless Networks

Cheng Zhan, Fuyuan Xiao, Li Tao, Siqi Chen

School of Computer and Information Science, Southwest University, China
{zhanc, xiaofuyuan, tli, siqichen}@swu.edu.cn

Abstract

There is an increasing demand for layered video streaming applications in wireless networks, which need data packets to be received strictly in-order and on time at the receivers. In this paper, we study efficient wireless broadcast scheduling problem in layered video streaming applications based on network coding, aiming at minimizing the number of packets which miss their deadlines. Using a weighted graph model, we formulate the problem as an integer linear programming and prove that it is NP-hard. We propose a packet encoding and broadcasting algorithm based on the maximum weight clique in the graph. Detailed analyses show that the appropriate settings of weight function can ensure that the packet with lower layer and earlier deadline is encoded and broadcast first. Simulation results show that our algorithm significantly reduces the deadline miss ratio in most cases, which is an important performance metric in layered video streaming applications.

Keywords: Broadcast scheduling, Layered video streaming, Network coding

1 Introduction

Broadcasting data to multiple users is widely used in several wireless applications, ranging from satellite communications to WiFi networks. By combining different source packets in a single coded packet, network coding can improve energy efficiency, throughput, and reduce delay over broadcast channels [1-3]. The works in [4-5] study coding based efficient broadcast schemes for loss recovery that allows instantaneous decoding. These coding schemes are also known as Instantly Decodable Network Codes (IDNC). Previous work on IDNC focuses on minimizing the completion time and decoding delay. However, there are few works considering the delay guarantee of data packets and in-order packet delivery, which is an important aspect of real time video streaming applications.

Recent development of commercial wireless services has created large scale demands for real time applications such as video streaming or interactive gaming. The emergence of new multimedia devices such as laptops or smartphones has motivated the research on efficient data delivery mechanisms that are able to adapt to dynamic network conditions and heterogeneous devices' capabilities [6-7]. The quality of experience of the different receivers depends on their display size, processing power, network bandwidth, etc. In order to accommodate for such a diversity, the data are progressively encoded in several quality layers [8-9], receivers can utilize the enhancement layer packet unless they have received all packets with the basic layer. This permits to offer a basic quality to receivers with limited capabilities, while other devices can have a higher quality of experience. For example, audio and video layered streaming applications, NetFlix and YouTube, need to play packets in-order and on-time in order to prevent interruption of the stream. The methods in existing studies on network coding cannot be used directly in multimedia applications owing to the characteristics of video streaming.

These real time layered video streaming applications have two distinct characteristics. First, the decoding of higher layer data depends on that of all the lower layers. The receiver cannot make use of layer k data unless it receives all the data from layer 1 to layer k. Second, the streaming packets have strict and urgent deadlines, a packet is useless (or less useful) after a short amount of time [10]. Therefore, it is desirable to design network coding schemes so that the needed video packets can be received in order before the deadline, which can contribute to improve the video quality. In this paper, we are interested in designing an efficient coding based broadcast scheme that minimizes the deadline miss ratio for video streaming under the condition with in-order packet delivery of different layers.

Consider a single hop broadcast scenario, similar to [11], a wireless sender delivers a streaming video consists of m layered packets to a set of n receivers. Each receiver is interested in receiving all the packets. From the prior transmissions, each receiver already has some packets and notifies (or informs) the sender using the feedback. Once the sender wants to transmit, it can

*Corresponding Author: Cheng Zhan; E-mail: zhanc@swu.edu.cn
DOI: 10.3966/160792642019012001005

encode based on the packets which receivers want and already have. See example shown in Figure 1, there are one bastion s and four receivers r_1, r_2, r_3, r_4. Suppose that s needs to transmit layered packets p_1, p_2, p_3 to the four receivers, a receiver cannot use p_3 unless it receives both p_1 and p_2. Each receiver already has some of the packets from prior transmissions, suppose that r_1 has p_2, r_2 and r_3 both have p_1, p_3, r_4 has p_1, p_2. s needs to send p_1, p_3 to r_1, send p_2 to r_2, r_3, send p_3 to r_1, r_4. Assume that the time taken for a packet transmission is 1 time slot, and the deadlines of p_2, p_3 are both 2 time slots. The deadline of p_1 is 1 time slot, since p_1 is the packet with the lower layer, it must be transmitted before the packets with higher layer.

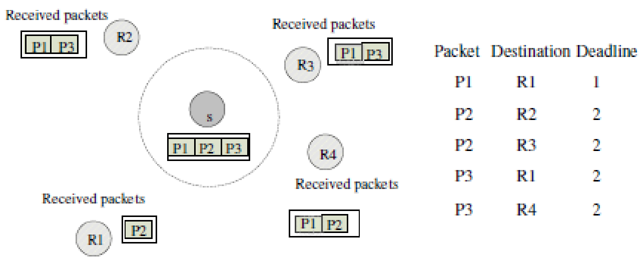


Figure 1. An example

The buffered packet information at the receivers is shown in Table 1, where 0 indicates that the corresponding packet is already received at the receivers and the non-zero element represents the deadline of the corresponding packet needed at the receivers. Without coding, s will transmit p_1, p_2, p_3 in sequence since the deadline of p_1 with the lowest layer is the smallest. Using this transmission strategy, there are two packets missing deadlines, the packet p_3 needed by r_1 and packet p_3 needed by r_4. According to the coding method introduced in [11], the sender will send an encoded packet to maximize the number of receivers which can decode out a needed packet, thus s will transmit $p_2 \oplus p_3$ first. r_1 can get p_3 since it already has p_2. Similarly, r_2 can get p_2, r_3 can get p_2, r_4 can get p_3. Although transmitting $p_2 \oplus p_3$ can make more receivers decode their wanted packets, some packets may still miss their deadlines. For example, the packet p_1 needed by r_1 will miss its deadline since the encoded packet transmitted at the first time slot does not contain p_1. Although r_1 can decode out p_3 from $p_2 \oplus p_3$, it cannot use p_2, p_3 without p_1 since p_1 is the packet with lowest layer. Intuitively, it will be better that the packets with lower layer are encoded and delivered earlier. Thus, if s transmits $p_1 \oplus p_2$ first, and then transmits p_3, p_1 will be decoded by r_1 in one time slot and within its deadline. Using this transmission strategy, there is no packet missing its deadline.

Table 1. Packet Information at Receivers

	$P1$	$P2$	$P3$
$R1$	1	0	2
$R2$	0	2	0
$R3$	0	2	0
$R4$	0	0	2

There are also some research works dealing with the broadcast problem multiple layers, which is similar to our work. In [12], the authors considered the broadcast problem that a set of packets forming the base layer has high priority and another set of packets forming the enhancement layers. However, the IDNC algorithms in [12] aimed to reduce the number of transmissions required for delivering all the packets, which is not suitable for video transmission. The authors in [13] discussed the hierarchical order of video layers and proposed a heuristic packet selection algorithm, aiming to balance between the number of transmissions for the base layer and all video layers. However, both works in [12-13] ignored the hard deadline and did not strictly consider the in order delivery of different layer packets before the deadline.

Inspired by the above limitations, this paper will deal with the efficient scheduling problem in wireless networks using network coding (xor coding) for layered streaming applications. We aim to minimize the number of layered packets which miss their deadlines. Our contributions are summarized as follows:

- We formulate the coding based efficient scheduling problem for the layered streaming applications as an integer linear programming and prove the problem is NP- hard.
- According to a weighted graph, we propose an encoding algorithm based on the maximum weight clique in the graph and analyze how to set the weight function to ensure that the packet with lower layer and earlier deadline be encoded and broadcast first.
- We compare the performance of the proposed encoding algorithm with some existing coding algorithms. Simulation results show that our method can significantly reduce the packet deadline miss ratio.

The remainder of this paper is organized as follows. The related work are described in Section 2. In Section 3, we will give the problem statement. In Section 4, we formulate the problem as an integer linear programming using a weighted graph model and prove the problem is NP- hard. The encoding algorithm based on maximum weight clique will be presented and the weight function setting is analyzed in Section 5. Simulation results are shown in Section 6. Finally, we conclude the paper in Section 7.

2 Related Work

Numerous network coding schemes for the broadcasting problem have been developed to meet different requirements of applications. There two categories coding schemes, one is random linear network coding (RLNC), the other is Instantly Decodable Network Codes (IDNC). IDNC is suitable for multimedia streaming due to its instant decodability property. However, the existing IDNC schemes did not deal with the in-order transmission which is common used in the layered video streaming. This motives our work that investigating the efficient coding scheme for layered video streaming transmission.

Windows based RLNC strategies for layered video transmission were adopted in [14-15]. In particular, a probabilistic approach for selecting coding windows was proposed in [14] where the coding windows can include the packets in the lower video layers into all coded packets to obtain high decoding probabilities for the lower layers. The work in [15] considered a scalable video transmission with a hard deadline and used a deterministic approach for selecting coding windows over all transmissions before the deadline. However, the encoding process of RLNC is complicated since there are many operations over large Galois fields performed in RLNC. In addition, the decoding process of RLNC needs complex matrix inversion which is not suitable for implementation in multimedia streaming.

IDNC has significant advantages for multimedia streaming due to its instant decodability property, which allows recovery of the video layers instantly when the receivers receive these packets, since the encoding and decoding process of IDNC is performed using simple XOR operations. Consider the point to multi-point (PMP) broadcast network, the authors in [2, 11] considered IDNC for wireless broadcast aiming at serving the maximum number of devices with any new packet. Moreover, the authors in [5, 16] formulated the problem of minimizing the number of time slots required for broadcasting into a stochastic shortest path framework. Several other works in IDNC considered different importance of packets in coding decisions [17-18]. In particular, IDNC for streaming transmission was adopted in [17] and the proposed IDNC schemes are asymptotically throughput optimal subject to deadline constraints. However, the aforementioned works developed IDNC schemes neither considered the relationship between different layers of source packets at the applications nor considered explicit packet delivery deadline.

The work in [11] studies the efficient coding schemes for real time application based on IDNC and finds a code that is instantly decodable by the maximum number of users. However, it does not consider the in-order packet delivery, which is a major characteristic of layered streaming applications. Joshi et al. [19] consider the problem of multicasting an ordered stream of packets to two users over independent erasure channels with instantaneous feedback to the source and provide a framework for analyzing in-order packet delivery in such applications, however they do not consider broadcast schedule strategy. Wang et al. [13] propose source packet combining schemes based on network coding for scalable video broadcast systems, aiming at reducing the number of retransmissions for decoding certain fraction of the source packets, however the packet deadline which is an important criteria in streaming applications is not considered. Different from the above work, our work design an efficient coding based broadcast scheme that minimize the deadline miss ratio over all receivers with the condition of in order packet delivery.

3 Problem Formulation

3.1 Layered Data

We consider a system that employs the layered video code extension to H.264/AVC video compression standard [20, 21]. A group of pictures in layered video has several video layers and the information bits of each video layer are divided into one or more packets. The video layers exhibit a hierarchical order such that each video layer can only be decoded after successfully receiving all the packets of this layer and its lower layers. The first video layer (known as the base layer) encodes the lowest temporal, spatial, and quality levels of the original video and the successor video layers (known as the enhancement layers) encode the difference between the video layers of higher temporal, spatial, and quality levels and the base layer. With the increase in the number of decoded video layers, the video quality improves at the receivers. Such an encoding enables receivers with limited capabilities to receive the base layer, while receivers with more resources decode the data in higher quality. The data from the base layer is the most important, followed by the data of the successive enhancement layers that offer incrementally finer levels of quality. A receiver can decode data from an enhancement layer only when it has decoded all the lower layers.

Suppose the video streaming data is progressively encoded in L layers. The data is further segmented in generations G_0, G_1,..., G_k, which are groups of time-constrained data (e.g., images in video case). We consider that the data of the l-th layer of a generation is packetized into α_l packets. Therefore, the first k layers consist of $\beta_k = \sum_1^k \alpha_k$ packets. Each generation has in total β_L packets, and every packet is associated with a delay deadline. For simplicity, we can arrange the data packets in a packet set $P = \{p_1, p_2, ..., p_m\}$, where packets are listed in increasing order of packet layer and m is the number of packets needed to be transmitted.

3.2 Network Model

Suppose a single hop wireless broadcast scenario which consists of a sender s and n receivers $r_1, r_2,\ldots, r_n$. s needs to transmit m layered packets $P = \{p_1, p_2, ..., p_m\}$ to the n receivers. When packet p_i is received at receivers, it can be successfully delivered to application if and only if packets $p_1, p_2, ..., p_{i-1}$ are received.

Receivers have already had some packets in their caches owing to the overhearing from prior transmissions. Each receiver needs a subset of packets in P. We use set $W(r_i)$ to denote the packets needed at receiver r_i and set $H(r_i)$ is used to denote the packets already had at receiver r_i, $W(r_i)\cap H(r_i)=\varnothing$, $W(r_i)\cup H(r_i)=P$. We assume that time is slotted, at each time slot the sender transmits one encoded packet. For $p_j \in W(r_j)$, $1\le j\le m$, we define a deadline T_{ij} to illustrate the delay threshold of packet p_j which r_i needs. If the deliver time of p_j to r_i is beyond T_{ij}, then p_j is useless to r_i. In other words, packet p_j needed by r_i misses its deadline. The packet deadline is related to delivering time but not receiving time, since the receiver can use the data with enhancement layer only when it has received all the lower layers. Thus, $T_{ij_1} < T_{ij_2}$, $j_1 < j_2$.

We define Deadline miss ratio as the percentage of packets that miss their deadlines:

$$Deadline\ miss\ ratio = \frac{S_m}{S_a}\times 100\%,$$

where S_m is the sum of numbers of packets missing deadlines at all receivers and S_a is the sum of numbers of packets needed by all receivers.

Our problem is that given the set of stored packets $H(r_i)$ at the receiver r_i, the set of packets $W(r_i)$ needed by the receiver r_i and the deadline T_{ij} of the packet p_j needed by receiver r_i, $1\le i\le n$, $1\le j\le m$, how to encode and transmit in each time slot to minimize the packet deadline miss ratio. In this paper, we only consider XOR coding instead of linear network coding since encoding and decoding operations using XOR is easy to be implemented with less overhead.

4 Layered Minimum Miss Ratio Problem

Given the information about the Have and Want sets of receivers, we form a weighted graph based on the IDNC graph in [5].

Definition 1: Given $R=\{r_1, r_2, \ldots, r_n\}$, $P=\{p_1, p_2,\ldots, p_m\}$, $W(r_i)\subseteq P, H(r_i)\subseteq P, W(r_i)\cap H(r_i)=\varnothing$, T_{ij}, $1\le i\le n$, $1\le j\le m$, we construct a weighted graph $G(V, E, t)$ as:

$$V=\{v_{ij}\mid \text{packet } p_j \text{ needed by } r_i\},$$
$$E=\{(v_{i_1j_1}, v_{i_2j_2})\mid j_1\ne j_2, p_{j_2}\in H(r_{i_1}), p_{j_1}\in H(r_{i_2})\}$$
$$\cup\{(v_{i_1j_1}, v_{i_1j_1})\mid j_1=j_2\}$$
$$t: V\rightarrow R+.$$

Table 2 shows the notations to be used in constructing the graph model and the proposed encoding algorithm. Figure 2 is the corresponding graph of the aforementioned example in section I. In Figure 2, v_{13} represents that r_1 needs packet p_3, $T_{13} = 2$ is the deadline of the needed packet p_3 by r_1. Since $p_3\in H(r_2)$ and $p_2\in H(r_1)$, there exists an edge (v_{13}, v_{22}). r_2 and r_3 need the same packet p_2, then there exists an edge (v_{22}, v_{32}).

Table 2. Notation

Symbol	Description
s	The sender
ri	Receiver i
n	Number of receivers
$H(ri)$	The set of packets already had at receiver ri
$W(ri)$	The set of packets needed at receiver ri
P	The set of packets to be transmitted
pj	The j-th packets to be transmitted
m	Number of packets to be transmitted
Tij	The deadline of packet pj needed by ri
G	The graph constructed based on the requests
$V(G)$	The vertex set of graph G
$E(G)$	The edge set of graph G
vij	A vertex corresponds to packet pj needed by ri in graph G
C	A clique in graph G

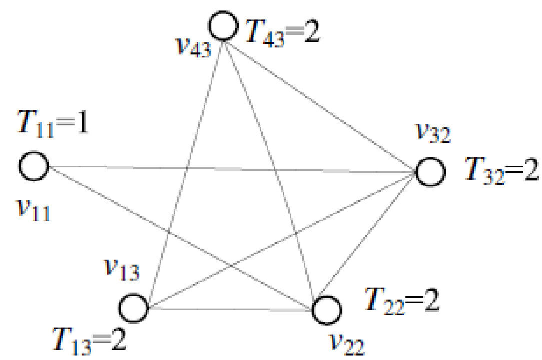


Figure 2. A weighted graph example

For each packet $p_j\in H(r_i)$, there is a corresponding vertex $v_{ij}\in V(G)$. t is a weight function, we denote $t(v_{ij}) = T_{ij}$. In other words, weight T_{ij} is assigned on vertex v_{ij} to represent the deadline of the needed packet p_j by r_i. It was shown in [5] that one encoded packet corresponding to a clique $C\subseteq V(G)$ can help some receivers recover their needed packets immediately.

Our objective is to decide how to encode and transmit the layered packets at every time slot to minimize the number of packets missing their deadlines. We need to find a clique partition $\{C_1, C_2, \ldots, C_k\}$ of the graph G, $1 \le k \le m$, from which we can transmit the encoded packet corresponding to C_l at time slot l, to minimize the number of packets which miss their deadlines. We refer this problem as the layered minimum miss ratio problem (LMMRP).

Theorem 1: The layered minimum miss ratio problem (LMMRP) is NP-hard.

Proof: Let us consider a special case of the layered minimum miss ratio aware clique partition problem. The packets in P are all belong to basic layer, and we set $T_i = 1$, $1 \le i \le n$. Thus we just need to find a maximum clique in the graph, the vertices which are not in the clique will miss their deadlines. This special case of the LMMRP problem is equivalent to finding a maximum clique in the graph, which is a well-known NP-Complete problem. Thus, the LMMRP problem is NP-hard.

Based on the graph model, we will give an integer linear programming formulation of the LMMRP problem. Assume that the current time is h, the corresponding clique is C_h, $1 \le h \le m$. Since T_{ij} is the deadline of packet corresponds to v_{ij}, if $h = T_{ij}$ and $v_{ij} \notin C_l$, $1 \le l \le h$, the packet corresponds to v_{ij} cannot be decoded from the encoded packet corresponding to $C_l (1 \le l \le h)$ before the deadline, which means that the packet corresponds to v_{ij} misses its deadline. Inspired by these observations, we can formulate the LMMRP problem as follows:

$$\textit{Minimize} \quad \sum_{h=1}^{m} \sum_{i=1}^{N} \delta_{ijh} y_{ijh}$$

$$\textit{Subject to} \quad \sum_{h=1}^{m} x_{ijh} = 1, \ 1 \le i \le n; 1 \le j \le m; \quad \textbf{(1)}$$

$$x_{i_1 j_1 h} + x_{i_2 j_2 h} \le 1, \ (v_{i_1 j_1}, v_{i_2 j_2}) \notin E(G) \quad 1 \le h \le m; \quad \textbf{(2)}$$

$$y_{ijh} = 1 - \sum_{k=1}^{h} x_{ijk}, \quad 1 \le i \le n; \quad 1 \le j, h \le m \quad \textbf{(3)}$$

$$x_{ij_2 h} \le \sum_{k=1}^{h=1} x_{ij_1 h}, \quad 1 \le j_1 \le j_2 \le m; \quad \textbf{(4)}$$

where

$$x_{ijh} = \begin{cases} 1 & \textit{if vertex } v_{ij} \textit{ is in clique } C_h, \\ 0 & \textit{otherwise} \end{cases}$$

x_{ijh} indicates whether vertex v_{ij} is selected in the clique C_h or not, $1 \le h \le m$. In the formulation, y_{ijh} is a slack variable which indicates that whether v_{ij} has been selected in one of the cliques $C_1, C_2, \ldots, C_h$. Indication variable δ_{ijh} is defined as follows, $\delta_{ijh} = 1$ if $T_{ij} = h$, otherwise $\delta_{ijh} = 0$. The constraint (1) means that the vertex v_{ij} can only be selected in one clique among $C_1, C_2, \ldots, C_m$, which guarantees that $C_1, C_2, \ldots, C_m$ is a clique partition of G. The constraint (2) represents that if $(v_{i_1 j_1}, v_{i_2 j_2}) \notin E(G)$, $v_{i_1 j_1}$ and $v_{i_2 j_2}$ cannot appear in the same clique. Constraints (1) and (2) are usually used in the formulation of the clique partition problem. In the constraint (3), if $y_{ijh} = 1$, v_{ij} has not been selected in one of the cliques $C_1, C_2, \ldots, C_h$. The constraint (4) means that if v_{ij_2} is selected in C_h, the vertex v_{ij_1} related to the basic layer packet must be selected in one of the cliques among $C_1, C_2, \ldots, C_{h-1}$, $1 \le j_1 \le j_2 \le m$. In other words, if the sender sends encoded packet consists of p_{j_2}, it must have sent an encoded packet consists of p_{j_1} before.

The objective function of the linear programming is the sum of the number of packets which miss their deadlines at all receivers. From the objective function, we can see that if $\delta_{ijh} = 1$ and $y_{ijh} = 1$, the packet corresponds to v_{ij} misses its deadline, that is because v_{ij} has not been selected in the cliques $C_1, C_2, \ldots, C_h$ and the packet corresponds to v_{ij} cannot be delivered by the receivers at time slot h.

Given the broadcast scenario in wireless network, we study the LMMRP problem using a weighted graph based on the needed packets, the packets already had at each receiver, and the deadlines of all packets. Based on the weighted graph, transmitting encoded packets to minimize deadline missed packets is equivalent to finding a clique partition solved by the integer linear programming mentioned above. However, solving the integer programming consumes much time when the graph is large, so we need to find an efficient algorithm to solve the LMMRP problem. In order to find a clique partition $C_1, C_2, \ldots, C_k$ of the graph G to minimize the number of packets which miss their deadlines, heuristics maximum weight clique algorithm can be used at every time slot, where every clique corresponds to an encoded packet.

5 Maximum Weight Clique Based Algorithm

In this section, we will propose an encoding algorithm for the LMMRP problem. The encoding algorithm is based on the maximum weight clique in the auxiliary graph mentioned in the above section, where the weight of the vertex is set as a decreasing function of the packet layer and packet deadline.

5.1 Encoding Algorithm

Some straightforward relationships between the deadline miss ratio and the properties of corresponding cliques can be observed as follows:

- Observation 1: finding a clique containing the vertex with lower layer first is beneficial for packets deliver with the enhancement layer.
- Observation 2: finding a clique with the maximum number of vertices can help serving more receivers at each time slot.
- Observation 3: finding the clique containing the vertex with the smallest deadline can serve the most urgent packets first.

The above observations indicate that besides paying attention to the number of packets which can be decoded at each time slot, higher priority for packets transmitting should be assigned to the packets which are belong to the low layer and have smaller deadlines. Thus, a clique finding algorithm should consider the above aspects.

A maximum weight clique in the graph accommodates both the weight and the number of vertices in the graph. Thus, we can assign weight $f(T_i)$ on vertex v_i and find a maximum weight clique in the graph, where $f(T_i)$ is a decreasing function of T_i. Since a packet with lower layer and smaller deadline should be encoded and transmitted earlier, $T_{ij_1} < T_{ij_2}$, $j_1 < j_2$, $f(T_i)$ should be a decreasing function of T_i which means that a vertex with larger weight corresponds to a packet with lower layer and smaller deadline. In the following subsection we will discuss how to set vertex weight function to meet the requirement in layered video streaming applications. A heuristic encoding algorithm is shown in Figure 3.

```
Input: G(V, E, t)
1. l ← 0;
2. while (V(G) ≠ ∅)
3.   Find a maximum weight clique C in G ;
4.   Denote the packet set in C as P_C = {p_c1, p_c2, ..., p_ck}
     p'_l ← p_c1 ⊕ p_c2 ⊕ ... ⊕ p_ck;
5.   l ← l + 1;
6.   Delete all vertices in C from G;
7. end while
Output: Encoded packet at time slot l: {p'_l}.
```

Figure 3. The encoding algorithm

Finding a maximum weight clique is NP-hard, it was shown in [22] that even for the maximum clique problem, no polynomial time algorithm can approximate the optimal solution within a factor of $|V|^{1-\epsilon}$ for any $\epsilon > 0$, we show our algorithm in Figure 4. In our algorithm, we maintain a working set U containing vertices which can construct a clique. We start from a vertex vij with the maximum f (T_{ij}) and set $U = \{v_{ij}\}$, then we consider the neighbors of v_{ij}, which is denoted by $N(v_{ij})$. We make a greedy choice, selecting $v_{k_1k_2}$ from the $N(v_{ij})$ with the maximum weight and add $v_{k_1k_2}$ to U. At step 8, $N(v_{ij})\backslash U$ means the set of all elements which are members of the $N(v_{ij})$, but not members of U. C_d is a vertex set contains the candidate vertices which might be put into working set U. At step 15, $\{\arg\max(f(T_{ij})), v_{ij} \in C_d\}$ } illustrates the vertices which have the largest weight. The time complexity of the algorithm for finding a maximum weight clique is $O(|V|^2)$.

```
Input: G(V, E, t)
1. max ← 0
2. for (v_ij ∈ G) do
3.   if (f(T_ij) > max) then
4.   max = f(T_ij)
5.   k1 ← i, k2 ← j
6. end for
7. U ← {v_k1k2}
8. while(N(v_k1k2) \ U ≠ ∅) do
9.   Cd ← ∅
10.  for(v_i1j1 ∈ N(v_k1k2) \ U)
11.      if ({v_i1j1} ⋃ U can construct a clique) then
12.          Cd ← Cd ∪ {v_i1j1}
13.      end if
14.  end for
15.  J ← {arg max(f(T_ij)), v_ij ∈ Cd}
16.  Randomly select v_ij ∈ J
17.  U ← U ∪ {v_ij}
18.  k1 ← i, k2 ← j
19. end while
Output: Maximum weight clique Cd
```

Figure 4. Maximum weight clique algorithm

5.2 Weight Function Analysis

The video streaming application need to deliver packets in-order and on-time in order to prevent interruption of the stream, thus the packet with lower layer and earliest deadline must be encoded and transmitted as early as possible. Our objective will be to make sure that the packets with lower layer and the smallest deadline is transmitted first according to finding a clique whose sum of deadline is the smallest with the lower layer, while maximizing the number of vertices in the clique at every time slot. In the following, we will discuss how to set the weight function of vertex to achieve the above objective by finding a maximum weight clique.

To transmit a packet with lower layer and smaller deadline earlier, we set $f(T_{ij}) = 2^{m-j}/T_{ij}$, where m is the number of packets the sender need to send. From the following theorem, we can see that using this weight function, the maximum weight clique is the one consisting of the vertex which corresponds to the packets with lower layer and small deadline. Thus, the lower layer packets that have the smaller deadline can be encoded and transmit earlier.

Theorem 2: The maximum weight clique consists of vertex which corresponds to the packets with lower layer and small $m-j$ deadline by setting $f(T_{ij}) = 2^{m-j}/T_{ij}$.

Proof: Assume that the maximum weight clique with weight function $f(T_{ij}) = 2^{m-j}/T_{ij}$ is $C = \{v_{i_1 j_1}, v_{i_2 j_2}, ..., v_{i_{k1} j_{k1}}\}$ listed in increasing order of packet layer, and $C' = \{v_{i'_1 j'_1}, v_{i'_2 j'_2}, ..., v_{i'_{k1} j'_{k1}}\}$ another clique listed in increasing order of packet layer, we prove the theorem with two cases.

Case 1: $j_1 < j'_1$. Suppose that C' layer has a large sum of weights than C, we prove the theorem by contradiction. Since $j_1 < j'_1$,

$$
\begin{aligned}
f(T_{i_1 j_1}) &= \frac{2^{m-j_1}}{T_{i_1 j_1}} > \sum_{k=j_1+1}^{m} \frac{2^{m-k}}{T_{i_1 j_1}} \\
&> \sum_{k=j'_1}^{m} \frac{2^{m-k}}{T_{i'_1 j'_1}} \\
&> \frac{2^{m-j'_1}}{T_{i'_1 j'_1}} + \frac{2^{m-j'_2}}{T_{i'_2 j'_2}} + ... + \frac{2^{m-j'_{k1}}}{T_{i'_{k1} j'_{k1}}} \\
&= f(T_{i'_1 j'_1}) + f(T_{i'_2 j'_2}) + ... + f(T_{i'_{k1} j'_{k1}})
\end{aligned}
$$

The first inequality holds since $2^{m-j_1} > \sum_{k=j_1+1}^{m} 2^{m-k}$. The second inequality holds because $T_{ij_1} < T_{ij_2}$, $j_1<j_2$. Since there are at most m needed packets, $|C'| < m$, where $|C'|$ is the cardinality of clique C', we get the last inequality. Thus the weight of clique C, $\sum f(T_{i_k j_k}) > \sum f(T_{i'_k j'_k})$, the weight of clique C'. This is a contradiction.

Case 2: $j_k < j'_k$ and $j_s < j'_s$ for $1 \le s < k$. We can get the same result for Case 2 with a similar analysis of Case 1 with minor modifications.

As a result, we get that the clique C is the maximum weight clique. Thus, the weight function of $f(T_{ij}) = 2^{m-j}/T_{ij}$ realizes finding the maximum weight clique with lower layer and earlier deadline. □

We will give a simple example of the application of our encoding algorithm in the following. Assume that there is a graph with four vertices v_1, v_2, v_3, v_4 and two edges (v_1, v_4),(v_2, v_3), and $T_1 = 2$, $T_2 = 3$, $T_3 = 3$, $T_4 = 8$. The packet corresponding to v_1 is base layer packet, and the packets corresponding to v_2, v_3, v_4 are all enhancement layer packets. Although $1/T_2 + 1/T_3 > 1/T_1 + 1/T_4$, the selected clique must be $\{v_1, v_4\}$ because the packet corresponds to vertex v_1 is the based layer packet and has earliest deadline. Using our weight function setting of the algorithm, the weight function values of $\{v_1, v_4\}$ and $\{v_2, v_3\}$ are $2^4/T_1 + 2/T_4$ and $2^3/T_2 + 2^2/T_3$ respectively. Since $2^4/T_1+2/T_4 > 2^3/T_2+2^2/T_3$, $\{v_1, v_4\}$ becomes the selected clique, which means that the packet corresponding to v_1 is sent first.

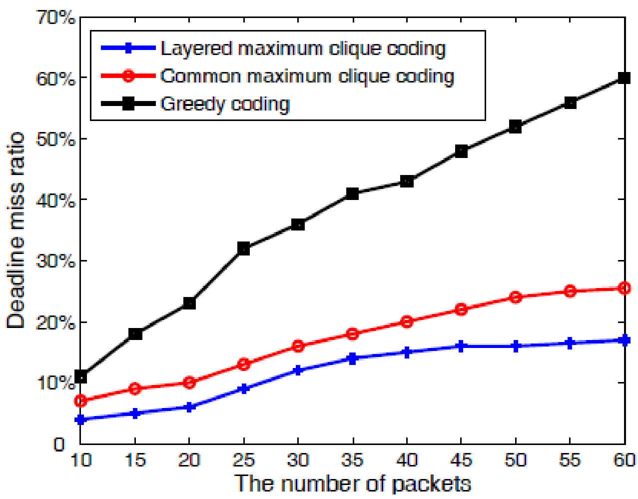
6 Simulations

In this section, we will present experimental performances of our coding based broadcast algorithm with layered packets, which have delay thresholds. The packets can be delivered to the application unless all the packets with lower layer are received.

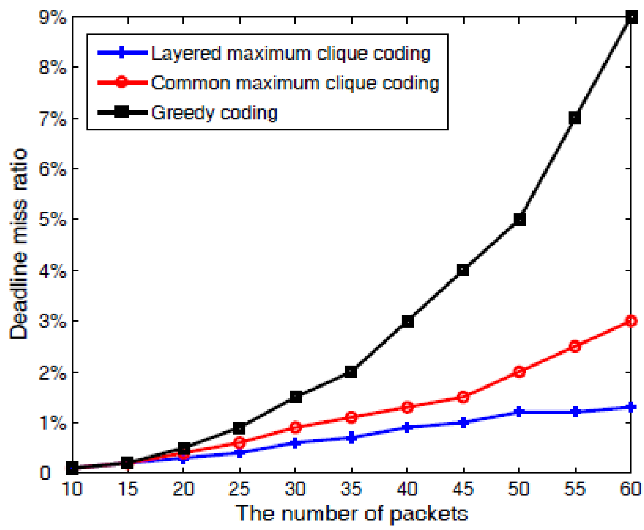
The simulation network topology is a broadcast scenario, where there is one sender and n receivers. Every receiver can receive the packets broadcast from sender, i.e., a base station and n users. All receivers are interested in the common file from the sender, thus we divide the file into m packets, and broadcast the m packets to all the receivers. According to the overhearing from the prior transmission, every receiver has already stored some packets. The stored packets and needed packets are randomly selected from the m packets with the same probability. The deadlines of packets are uniformly distributed in [*DMIN*, *DMAX*] with the condition that $T_{ij_1} < T_{ij_2}$, $j_1<j_2$. The packet deadline is related to deliver time but not receiving time, since the receiver can use data from an enhancement layer only when it has received all the lower layers. We use the deadline miss ratio as performance metric, which is an important performance metric in the layered video streaming applications.

We present the simulation results comparing the performance of our proposed algorithms to the following algorithms. *Greedy coding algorithm*, which also consider the delay when coding as mentioned in [11], uses IDNC strategies to serve maximum number of receivers in each transmission while ignoring in-order delivery and the deadline of different packets. *Common maximum clique coding algorithm*, which uses IDNC strategies to reduce the number of transmissions required for minimizing deadline miss ratio while ignoring in-order delivery of packet layers, sets $f(T_{ij}) = 1/T_{ij}$, It only considers the deadline. For each simulation setting, we present the average performance of 200 runs.

Figure. 5 shows the network coding gain in terms of deadline miss ratio of different algorithms for different deadlines settings and different numbers of packets for (n = 5, *DMIN* = 1). Deadline miss ratio is defined as the percentage of packets that miss their deadlines. We set *DMAX* = 20 in Figure 5(a) and *DMAX* = 40 in Figure 5(b). In Figure 5 we can see that our layered maximum clique coding algorithm performs better than common maximum clique coding which consistently performs better than greedy algorithm. Our algorithm can reduce the deadline miss ratio about 60% on average compared to greedy coding algorithm. The reason is that our algorithms consider the effect of packet layer and deadline and try to broadcast the packets with smaller layer and deadline first. As shown in Figure 5, with the increase of *m*, the deadline miss ratio increases. When the number of packets *m* is larger, the network coding gain is larger. The reason is that the deadline was randomly selected in [*DMIN*, *DMAX*] and with the number of packets increasing there are more packets need to be received by the receivers, the packet miss ratio increases.



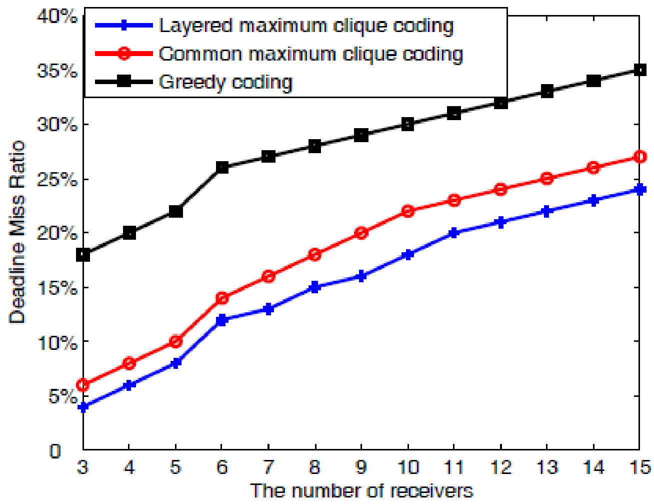
(*a*) Deadline in [1, 20]



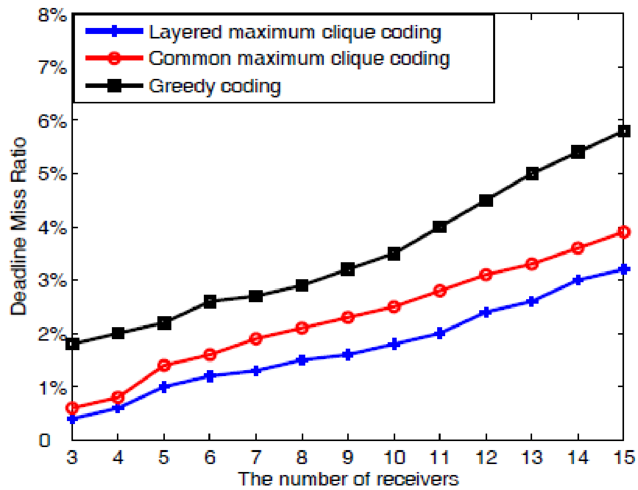
(*b*) Deadline in [1, 40]

Figure 5. Deadline miss ratio V.S. *m*

Figure 6 shows the network coding gain in terms of deadline miss ratio of different algorithms for different deadlines settings and different numbers of packets for (m = 20, *DMIN* = 1). As expected, with the increase of the number of receivers, the deadline miss ratio is increasing. Our algorithm can reduce the deadline miss ratio about 50% on average compared to greedy coding algorithm in this settings. The reason is that there are more needed packets being set at the same deadlines. From Figure 6 we can also see that the deadline miss ratio is smaller when *DMAX* = 40 compared to *DMAX* = 20, it is easy to see when the packet deadline is large, there will be less packets missing their deadlines.



(*a*) Deadline in [1, 20]



(*b*) Deadline in [1, 40]

Figure 6. Deadline miss ratio V.S. *n*

7 Conclusion

In this paper, we focus on network coding based efficient broadcast scheduling for wireless layered video streaming application using a weighted graph model and aim at minimizing the number of packets which miss their deadlines. We prove the layered minimum miss ratio problem is NP-hard and propose an integer linear programming formulation. By assigning vertex weight considering both the packet layer and packet deadline, we propose an encoding algorithm based on the maximum weight clique in the graph. Simulation results show that our algorithms can reduce the deadline miss ratio， which is an important performance metric in the layered video streaming applications.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61702426, No. 61602391), and Fundamental Research Funds for the Central Universities (No. XDJK2015C104, No. SWU115002).

References

[1] C. Fragouli, J. Widmer, J. Le Boudec, Efficient Broadcasting Using Network Coding, *IEEE/ACM Transactions on Networking*, Vol. 16, No. 2, pp. 450-463, April, 2008.

[2] P. S. Pandey, N. Purohit, S. Verma, Wireless Networks Using Matroid along with Network Coding, *Journal of Internet Technology*, Vol. 18, No. 4, pp. 775-786, July, 2017.

[3] W.-D. Fang, L.-H. Shan, G.-Q. Jia, X.-H. Ji, S.-J. Chen, A Low Complexity Secure Network Coding in Wireless Sensor Network, *Journal of Internet Technology*, Vol. 17, No. 5, pp. 905-913, September, 2016.

[4] S. Sorour, S. Valaee, Completion Delay Minimization for Instantly Decodable Network Coding with Limited Feedback, *Proc. of IEEE International Conference on Communications (ICC 2011*), Kyoto, Japan, 2011, pp. 1-5.

[5] N. Aboutorab, P. Sadeghi, S. Sorour, Enabling a Tradeoff between Completion Time and Decoding Delay in Instantly Decodable Network Coded Systems, *IEEE Transactions on Communications*, Vol. 62, No. 4, pp. 1296-1309, April, 2014.

[6] R.-Y. Chang, S.-L. Peng, G. Lee, C.-J. Chang, Comparing Group Characteristics to Explain Community Structures in Social Media Networks, *Journal of Internet Technology*, Vol. 16, No. 6, pp. 957-962, November, 2015.

[7] S. Lakshminarayana, A. Eryilmaz, Multirate Multicasting With Intralayer Network Coding, *IEEE/ACM Transactions on Networking*, Vol. 21, No. 4, pp. 1256-1269, August, 2013.

[8] K.-C. Tsai, G.-L. Li, M.-J. Chen, C.-H. Yeh, A Selection Mechanism for Multiple Error Concealment Techniques in Scalable Video Coding, *Journal of Internet Technology*, Vol. 18, No.5, pp. 1031-1041, September, 2017.

[9] J.-S. Chiang, C.-H. Hsia, T.-H. Hwang, An Efficient VLSI Architecture of Embedded Sub-Band Coding with Optimal Truncation for Scalable Video Coding, *Journal of Internet Technology*, Vol. 18, No. 4, pp. 705-711, July, 2017.

[10] K. S. Kim, C.-p. Li, E. Modiano, Scheduling Multicast Traffic with Deadlines in Wireless Networks, *Proc. of IEEE INFOCOM* 2014-*IEEE Conference on Computer Communications*, Toronto, ON, Canada, 2014, pp. 2193-2201.

[11] A. Le, A. S. Tehrani, A. G. Dimakis, A. Markopoulou, Instantly Decodable Network Codes for Real-time Applications, *Proc. of International Symposium on Network Coding* (NetCod), Calgary, AB, Canada, 2013, pp. 1-6.

[12] M. Muhammad, M. Berioli, G. Liva, G. Giambene, Instantly Decodable Network Coding Protocols with Unequal Error Protection, *IEEE International Conference on Communications* (ICC), Budapest, Hungary, 2013, pp. 5120-5125.

[13] S. Wang, C. Gong, X. Wang, M. Liang, Instantly Decodable Network Coding Schemes for In-order Progressive Retransmission, *IEEE Communications Letters*, Vol. 17, No. 6, pp. 1069-1072, June, 2013.

[14] D. Vukobratovic, V. Stankovic, Unequal Error Protection Random Linear Coding Strategies for Erasure Channels, *IEEE Transactions on Communications*, Vol. 60, No. 5, pp. 1243-1252, May, 2012.

[15] S. Wang, C. Gong, X. Wang, M. Liang, An Efficient Retransmission Strategy for Wireless Scalable Video Multicast, *IEEE Wireless Communications Letters*, Vol. 1, No. 6, pp. 581-584, December, 2012.

[16] S Sorour, S Valaee, Completion Delay Minimization for Instantly Decodable Network Codes, *IEEE/ACM Transactions on Networking (TON)*, Vol. 23, No. 5, pp. 1553-1567, October, 2015.

[17] X. Li, C.-C. Wang, X. Lin, On the Capacity of Immediately Decodable Coding Schemes for Wireless Stored-video Broadcast with Hard Deadline Constraints, *IEEE Journal Selected Areas in Communications*, Vol. 29, No. 5, pp. 1094-1105, May, 2011.

[18] M. S. Karim, P. Sadeghi, N. Aboutorab, S. Sorour, In Order Packet Delivery in Instantly Decodable Network Coded Systems over Wireless Broadcast, *Proc. of IEEE International Symposium on Network Coding* (NetCod), Sydney, NSW, Australia, 2015, pp.11-15.

[19] G. Joshi, Y. Kochman, G. W. Wornell, Throughput-smoothness Trade-offs in Multicasting of an Ordered Packet Stream, *Proc. of International Symposium on Network Coding* (NetCod), Aalborg, Denmark, 2014, pp. 1-6.

[20] P. Seeling, M. Reisslein, B. Kulapala, Network Performance Evaluation Using Frame Size and Quality Traces of Single-layer and Two-layer Video: A Tutorial, *IEEE Communications Surveys &Tutorials*, Vol. 6, No. 3, pp. 58-78, Third Quarter, 2004.

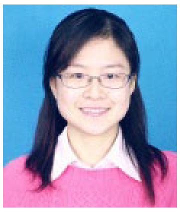
[21] H. Schwarz, D. Marpe, T. Wiegand, Overview of the Scalable Video Coding Extension of the h. 264/avc Standard, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, No. 9, pp. 1103-1120, September, 2007.

[22] J. Ha˚stad, Clique is Hard to Approximate within n1−ε, *Proc. of 37th Annual IEEE Symposium on the Foundations of Computer Science* (FOCS), Burlington, VT, 1996, pp. 627-636.

Biographies

Cheng Zhan received the Ph.D. degree in computer science from the University of Science and Technology of China in 2011. He is currently in the School of Computer and Information Science, Southwest University, China. His research interests include network coding, wireless network optimization, multimedia transmission and distributed storage.

Fuyuan Xiao was born in Guangxi Province, China. She received the Ph.D. degree from the Kumamoto University, Japan, in 2014. She is currently in the School of Computer and Information Science, Southwest University. Her research interests include optimization algorithm, data analysis, and vehicular network.



Li Tao received the Ph.D. degree in computer science from the Hong Kong Baptist University in 2014. She is currently in the School of Computer and Information Science, Southwest University, China. Her research interests include computational and complex network analysis.

Siqi Chen was born in Chongqing, China. He received the Ph.D. degree from the Maastricht University, in 2014. He is currently in the School of Computer and Information Science, Southwest University. His research interests include optimization algorithm, game theory based optimization.