Optimal Scheduling with Pairwise Coding under Heterogeneous Delay Constraints

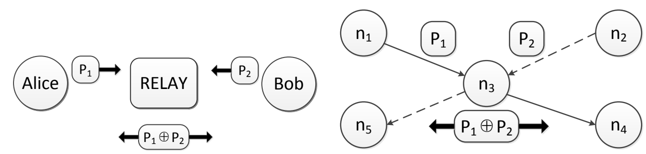
Yafei Mao*, Chao Dong†, Haipeng Dai*, Xiaojun Zhu*, and Guihai Chen*‡§
*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, CHINA
†College of Communications Engineering, P.L.A University of Science and Technology, Nanjing 210007, CHINA
‡Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, CHINA
§ Corresponding author, gchen@nju.edu.cn

***Abstract*—Network coding has the potential to provide powerful support to transmit real-time traffic in wireless network. In this paper, we utilize pairwise coding to schedule the flows which have heterogeneous delay constraints and weights. Our goal is to maximize the weighted sum of scheduled packets that satisfy the delay constraints. We formulate the problem as an integer linear programming problem, and propose two algorithms to solve it. The first algorithm drops the integral constraints and rounds the fractional solutions in such a way that the rounded solution is also optimal. Inspired by the first algorithm and for better running time, we propose the second algorithm based on a minimum cost flow formulation. The formulation is proved to be equivalent to the original integer linear programming formulation. Simulations are conducted to show the effectiveness of our approach over two greedy algorithms.**

I. Introduction

With the tremendous advance of broadband wireless technologies, the demand of supporting real-time multimedia applications has become increasingly important in wireless network [1]. Video-on-demand, battlefield situation information, online interactive gaming and meeting are typical real-world applications which specify stringent Quality of Service (QoS) requirements. In particular, the flows in these applications typically have hard and, more often than not, heterogenous delay constraints. However, the time-varying nature of wireless channel [2] gives rise to great challenges in supporting these real-time applications. To address this issue, network coding has been proposed and studied in the past few years, which is proved to be able to considerably improve throughput and reliability in wireless networks. Because of its huge advantages, this technique has found a wide range of applications in real-time traffic delivery (e.g., [3]–[5]).

Currently, there have emerged two primary kinds of network coding schemes, i.e., inter-session network coding and intra-session network coding. Rather than mixing packets from the same flow in intra-session network coding, inter-session network coding needs to encode packets from multiple flows. In this paper, we focus on inter-session network coding. One basic scheme of inter-session network coding is shown in Fig. 1(a). There are two nodes, say Alice and Bob, exchanging packets P_1 and P_2 through a relay node. The relay node performs XOR operation and broadcasts $P_1 \oplus P_2$ to both Alice and Bob.



(a) Reverse carpooling topology (b) X topology

Fig. 1. Inter-session pairwise network coding

Alice and Bob can obtain wanted packet by XOR-ing $P_1 \oplus P_2$ with their own packet. Formally, this topology is called reverse carpooling [6]. The second topology incorporates opportunistic listening technique, generalizing the reverse carpooling to X topology as shown in Fig.1(b). Under X topology, node n_5 can obtain the wanted packet P_2 through XOR-ing packet $P_1 \oplus P_2$ with packet P_1 which can be overheard from node n_1. Under both topologies, the throughput can be improved with pairwise coding which encodes two packets from a pair of flows at one time.

References [5] [6] [7] studied the scheduling of inter-session coding flows to achieve a balance between throughput and delay requirement, but none of them considered heterogenous delay constraints, which, however, are practical concerns for real-time application scheduling problem [8]–[10].

In this paper, we study the problem of Optimal Pairwise Coding Scheduling (OPCS) in wireless network. The flows have pairwise coding opportunities, heterogeneous delay constraints and packet weights considering flow priority. Our goal is to maximize the weighted sum of packets which successfully scheduled within their delay constraints. The contributions of our work are as follows:

1) We present an analytical optimization model for scheduling flows with pairwise coding opportunities, heterogeneous delay constraints and weights. In our settings for this model, the relay can perform pairwise coding on any pair of flows satisfying reverse carpooling topology or X topology, and the flows may have random arrival patterns, various delay constraints and different weights.
2) We formulate the problem as Integer Linear Programming (OPCS-ILP) and drop the integral constraints to construct a Linear Programming problem (OPCS-LP). We prove that any optimal solution to OPCS-LP can be equivalently transformed to an optimal solution to OPCS-ILP within polynomial time.

3) We further propose a more efficient optimal solution based on Minimum Cost Flow (OPCS-MCF) by uncovering the relationships among coding flows.
4) Finally, we compare our optimal solution with other two algorithms, Maximum Weight Packet First (MWPF) [5] and Urgent Packet First (UPF).

The remainder of the paper is organized as follows. Section II introduces related works. We formulate the optimization framework in Section III and provide two optimal solutions in Section IV. Experimental results and conclusion are presented in Sections V and VI, respectively.

II. Related Works

While abundant research works such as [8], [9] and [10] have been devoted to scheduling real-time traffic with heterogenous delay requirements over wireless network, the results may not be directly applied to coding flows. One major challenge for scheduling coding traffic is that coding schemes make the dependence among flows more complicated. References [3] [11] studied the scheduling with intra-session network coding. Yang *et al.* [3] focused on the scenario of broadcasting packets at access point (AP) and provided a scheduling policy which strikes a balance between maximizing throughput and minimizing packets delay through the adjustment of coding blocks. As for works on inter-session network coding (e.g., [5] [12] [7] [6]), Seferoglu *et al.* [5] studied the scheduling of video streaming over wireless network, and proposed a heuristic approach to improve both video quality and throughput. An end-to-end delay analysis of coding flows was first provided in [7]. However, these works did not consider heterogeneous hard delay constraints of inter-session coding flows.

III. System model

A. Network Model

We consider a wireless network operating a time division multiplexing scheme, and focus on a single relay to forward flows from various sources to destinations. We apply a discrete time model where time is slotted and numbered as $1, 2, 3, \ldots$ [13]–[15]. The relay node can make exactly one transmission in a time slot and the duration of a time slot is set to be the time for broadcasting one packet. Besides that, a set of T consecutive slots is called an interval. Let $\Gamma \overset{def}{=} \{1, ..., T\}$. We assume that there are N flows $1, 2, \ldots, N$ passing relay node R during an interval Γ, creating N virtual queues $q \in \{1, 2, ...N\}$. The length of q_i is l_i at the beginning of Γ. For packets arriving later, they are buffered at relay and are not involved in current round of scheduling.

Let T_i ($T_i \leq T$) be the delay requirement of packets in flow f_i so that packets in q_i should be scheduled before time T_i. Without loss of generality, we assume that the queues are arranged in an increasing order of delay requirements such that $T_1 \leq T_2 \leq \cdots \leq T_N$. The weight w_i of packets in f_i is decided by the priority of f_i. We assume the number of scheduled packets from flow f_i during an interval is n_i, then we define the utility U as

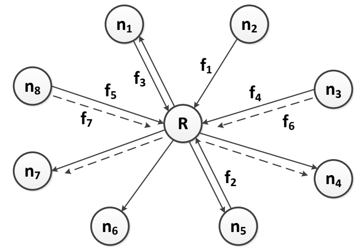


Fig. 2. Illustration of network model

$$U = \sum_i w_i \cdot n_i.$$

The relay node can perform pairwise coding on any pair of flows if they satisfy reverse carpooling topology in Fig. 1(a) or X topology in Fig. 1(b). Under the considered schemes, each flow can code with at most one flow, and we refer to them as a coding pair. The coding packet can be sent in only one time slot while containing two native packets. Our objective is to maximize the total utility of flows with different delay constraints and weights, where the flows have pairwise coding opportunities among them. These two topologies are quite typical in ad-hoc network and mesh network, so the relay node can find many pairwise coding opportunities on flows passing through it. Additionally, Immediately-Decodable intersession Network Coding has substantially smaller decoding delay, and incurs much lower encoding complexity [16]. It is both practical and reasonable for us to consider pairwise coding scheme.

We illustrate our model in Fig. 2. There are five unicast flows passing through relay node R. f_1 from n_2 to n_6 is a single flow which has no coding opportunity, f_2 from n_5 to n_1 and f_3 from n_1 to n_5 satisfy reverse carpooling and constitute a coding pair. f_4 from n_3 to n_7 and f_5 from n_8 to n_4 satisfy X topology and constitute another coding pair. Note that not all packets in the flow pairs have coding opportunities due to channel error in X topology. Nevertheless,

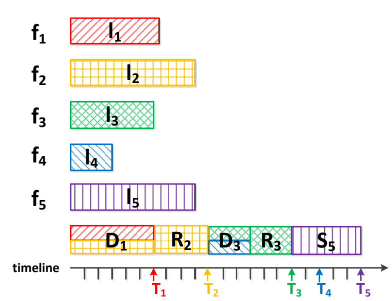


Fig. 3. Illustration of standard schedule

we can identify these packets by reception report or delivery probability estimation introduced in [17]. In this case, the packets in a coding pair which can not be used in coding will be labeled as two new single flows.

As for the flows f_4 and f_5 in Fig. 2, the packets in f_4 which do not have coding opportunity constitute a new single flow called f_6, and the packets in f_5 which do not have coding opportunity constitute another new single flow called f_7. We depict them by dashed line in Fig. 2.

B. Problem Formulation

We define two sets for flows passing through relay node, set $\mathcal{P}_s$ consists of single flows which do not have coding opportunity, set $\mathcal{P}_c$ includes flow pairs with coding opportunity. In the network model illustrated in Fig. 2, we have $\mathcal{P}_s = \{1, 6, 7\}$ and $\mathcal{P}_c = \{(2, 3), (4, 5)\}$. For single flow $k \in \mathcal{P}_s$, we refer to the packets that are successfully transmitted in f_k as single packets S_k. As for coding pair $(i, j) \in \mathcal{P}_c$ with $i < j$, we refer to the encoding packets for two flows that are successfully transmitted as double packets D_i. For packets which may be sent alone without coding in f_i or f_j, we refer to them as residual packets R_i and R_j respectively. By our definition, we can further deduce that only one flow in (i, j) has residual packets, i.e., $R_i \cdot R_j = 0$, otherwise the residual packets from both sides can be coded together in continue. Now we define an standard form of solution in our Optimal Pairwise Coding Scheduling (OPCS) problem for better illustration.

Definition 1: (**standard schedule**). A standard schedule for OPCS is an ordered sequence of packets sent during an interval. Packets from the same flow is sent consecutively, and all packets in the scheduling are arranged in an increasing order of delay requirements.

That is to say the packets are ordered in consecutive blocks of D, R, S from f_1 to f_N in a standard schedule. Note that the coding packet has the same delay constraint as the flow with smaller delay constraint in a coding pair.

Theorem 1: Given any feasible schedule during an interval, we can transform it into an equivalent standard schedule.

Proof: For any feasible schedule, consider two consecutive packets in this schedule. If they are from different flows, we can move the one that has shorter delay constraint ahead of the other. Then the delay constraints of both packets can still be satisfied, and the utility remains the same. Repeating this operation transforms the schedule to be a standard schedule. ■

We use an example in Fig. 3 to illustrate standard schedule by D, R, S. There are five flows f_1, f_2, f_3, f_4, f_5 passing through relay node R. (f_1, f_2) and (f_3, f_4) are two couples of coding pair, and f_5 is a single flow. It holds that $T_1 < T_2 < T_3 < T_4 < T_5$ and $l_1 < l_2, l_4 < l_3$. In this case, we send D_1 coding packets from both f_1 and f_2, and there are R_2 packets sent alone in f_2. Send D_3 coding packets of coding pair (f_3, f_4), and residual packets R_3 from f_3. We have no packets sent alone in f_4. As for single flow f_5, we send S_5 single packets.

We can formulate OPCS using the definitions of D, R, S as an Integer Linear Programming problem called OPCS-ILP.

$$\max_{D,R,S} \sum_{(i,j)\in\mathcal{P}_c} ((w_i + w_j) \cdot D_i + w_i R_i + w_j R_j) + \sum_{k\in\mathcal{P}_s} w_k S_k$$

TABLE I

NOTATIONS

Symbol	Meaning
Γ	Interval
T	Length of an interval
R	Relay node in charge
f_i	ith flow at R
q_i	virtual queue for f_i at R
T_i	Delay requirement of packets in f_i
l_i	The length of q_i
w_i	Weight of packets in f_i
D_i	Coding packets in coding pair $(i, j), i < j$
R_i	Residual packets in f_i
S_i	Single packets in f_i
$\mathcal{P}_s$	Set of single flows
$\mathcal{P}_c$	Set of coding pairs

s.t.

$$S_k \leq l_k,\ \forall\, k \in \mathcal{P}_s \tag{1}$$

$$D_i + R_i \leq l_i\ \wedge\ D_i + R_j \leq l_j,\ \forall\, (i,j) \in \mathcal{P}_c \tag{2}$$

$$\sum_{\substack{i\leq t\\(i,j)\in\mathcal{P}_c}} (D_i + R_i) + \sum_{\substack{j\leq t\\(i,j)\in\mathcal{P}_c}} R_j + \sum_{\substack{k\leq t\\k\in\mathcal{P}_s}} S_k \leq T_t,\ \forall\, t \leq N \tag{3}$$

$$D_i, S_i, R_i \in \{0, 1, 2, 3, ...\}. \tag{4}$$

Constraint (1) says that packets being scheduled in single flows cannot exceed the length of flow. Constraint (2) is the length constraints for coding pairs. Constraint (3) specifies that scheduling packets shouldn't exceed their delays. Constraints (4) indicates that D, R, S must be nonnegative integers.

For convenience, the notations used are summarized in Table I.

IV. OPTIMAL SOLUTIONS

In this section, we drop integral constraints in OPCS-ILP and construct a new problem based on Linear Programming which is called OPCS-LP. We prove that given an optimal solution to OPCS-LP, an equivalent optimal solution to OPCS-ILP can be constructed in polynomial time. Another optimal solution based on Minimum Cost Flow (OPCS-MCF) is then provided. We verify that OPCS-MCF has a better time complexity than OPCS-LP.

A. Optimal solution based on LP

In this subsection, we drop integral constraints in OPCS-ILP and reformulate it as OPCS-LP, then we prove that OPCS-LP can provide integral optimal solution to OPCS. By taking an in-depth study into the structural properties of OPCS, we conclude that there must be an integral optimal solution by OPCS-LP and all fractional optimal solutions can be rounded to it.

Given an optimal solution to OPCS-LP, if all variables D, R, S are integers, the solution is already an integral optimal solution. For fractional solutions, we have the following theorem.

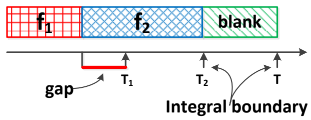


Fig. 4. Illustration of integral boundary and gap

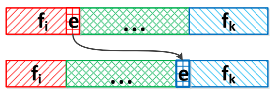


Fig. 5. Illustration of tiny exchange

***Theorem** 2:* Given a fractional optimal solution to OPCS-LP, we can round it to an equivalent integer solution in polynomial time.

Proof: First we divide total interval Γ by integral boundaries and cut Γ into several partitions.

***Definition** 2:* (**integral boundary**). An integral boundary is certain deadline T_i that there is no time gap between the last scheduling time of f_i and T_i, the scheduling of f_i is exactly stopped by its deadline constraints.

We further illustrate the notations of integral boundary and time gap in Fig. 4. Without loss of generality, we refer to the blank time slots at the end of standard schedule as packets from a long single flow, whose weight is 0 and deadline is T, then T is an integral boundary itself. We define a manipulation on fractional packets called "tiny exchange", which is widely used in our rounding process.

***Definition** 3:* (**tiny exchange**). Tiny exchange from fractional packets f_i to f_k is to cut a fractional part e from f_i, and add it to f_k without breaking any constraint. This is illustrated in Fig. 5.

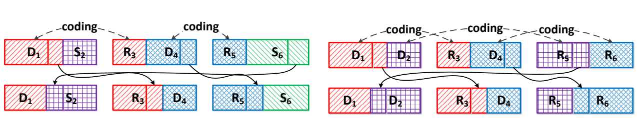
We start our rounding from single packets, and propose some observations on these fractional packets.

***Observation** 1:* For fractional single packets which distribute in one integral partition, they must be of same weight.

***Observation** 2:* Fractional D, R packets from the same coding pair can not coexist in one integral partition.

For a coding pair (i, j), if only the D packets or the R packets are fractional, or both D, R packets are fractional but satisfy $D_i + R_j < l_j$, we can view these packets as fractional single packets which must be of same weight of other factional single packets within a partition. Because the numbers of packets sent in these flows are less than their length and we can perform tiny exchange on them within each partition as we proved before. We can round these fractional packets until they become integers or the D, R packets satisfy $D_i + R_j = l_j$.

After these rounding processes, there is at most one fractional single flow within each partition. Fractional D_i, R_j packets from coding pair (i, j) distribute in different partitions and satisfy $D_i + R_j = l_j$. Partitions with fractional packets are classified as two types: with a fractional single flow or without fractional single flow, as shown in Fig. 6. Also known is that the amount of fractional single flows during interval Γ is either 0 or more than 2, because the sum of fractional single flows must be an integer. The remained fractional flows can also be rounded into integer ones. We omitted the detailed proof here due to page limitation. ∎



(a) Partitions with one fractional single flow

(b) Partitions without fractional single flow

Fig. 6. Tiny exchange on coding pairs

TABLE II
TRANSFORMATION RULES

Rules	Description
a)	Put two virtual vertices in set V: V_S (the source vertex) and V_T (the sink vertex).
b)	We cut interval Γ into several parts by delay constraints of flows, each part is represented as a vertex V_k^t, called interval vertex.
c)	For a single flow f_k without coding opportunity, insert vertex V_k^s to V, called single vertex.
d)	For coding pair (i, j), insert vertices V_i^c and V_j^c to V as coding vertex, vertex V_i^r is call residual vertex for f_i.
e)	Edges between source vertex and single vertices: insert an edge (V_S, V_k^s) for each single vertex V_k^s. The capacity on this edge is l_k, and the cost is 0.
f)	Edges between single vertices and interval vertices: insert an edge (V_k^s, V_u^t) to E for every $u \leq k$. The capacity of this edge is l_k and cost is $-w_k$.
g)	Edges between source vertex and coding pair vertices are more sophisticated. For coding pair (i, j), we insert two edges from source (V_S, V_i^r) and (V_S, V_j^c). The capacities are l_i and l_j, respectively, and the costs are 0.
h)	Edges between coding pair vertices: add edge (V_j^c, V_i^r) to E, which represents coding packets. The capacity on this edge is l_j, and the cost is $-w_j$. Insert (V_i^r, V_i^c) which represents the all the packets sent from f_i. The capacity is l_i, and the cost is 0.
i)	Edges between coding pair vertices and interval vertices: insert edge (V_i^c, V_u^t) for every $u \leq i$, the capacity on this edge is l_i and the cost is $-w_i$. Process for adding (V_j^c, V_u^t) is quite similar.
j)	Edges between interval vertices and sink vertex: insert edge (V_u^t, V_T) to E for every interval vertex. The capacity on (V_u^t, V_T) is $T_u - T_{u-1}$, this is the number of time slots within this interval part.

B. Optimal solution based on MCF

In this section, we show that OPCS can be equivalently transformed into a Minimum Cost Flow (MCF) problem which allows a lower time complexity. The MCF is to determine a least cost shipment of a commodity through a network in order to satisfy demands at certain nodes from available supplies at other nodes [18]. Although MCF is commonly used in scheduling optimization, it is not easy to apply OPCS directly into a MCF problem. This is because we need to consider both heterogenous delay constraints and dependence among coding flows which make the formulation more complex.

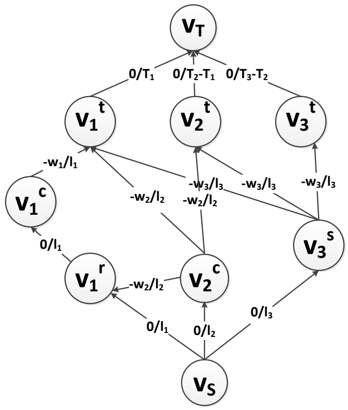


Fig. 7. Minimum cost maximum flow model

We perform our transformation according to the rules in Table. II, where rules $a) \sim d)$ and $e) \sim j)$ respectively give the vertices and edges meanings. The costs on edges are defined as either zeros or negative of weights, however most minimum cost flow algorithms support edges with negative costs. Define set vertex set V and edge set E for OPCS-MCF graph.

We take an example to illustrate how to transform OPCS into a corresponding MCF by applying these rules. Suppose there are three flows f_1, f_2, f_3 passing through our relay node, $\{f_1, f_2\}$ is a coding pair, f_3 is a single flow. The corresponding MCF graph is constructed in Fig. 7. As for coding pair (f_1, f_2), the flow on edge (V_s, V_1^r) stands for residual packet R_1, and flow on edge (V_2^c, V_1^r) represents coding packets D_1, thus flow on edge (V_1^c, v_1^t) is the number of f_1 packets which are sent before the deadline T_1. Flow on edge (V_s, V_2^c) stands for all packets sent in f_2, flows on edges (V_2^c, V_1^t) and (V_2^c, V_2^t) represent the residual packets R_2 sent during time interval $[1, T_1]$ and $[T_1+1, T_2]$ respectively. As for single flow f_3, flows on edges (V_3^s, V_1^t), (V_3^s, V_2^t) and (V_3^s, V_3^t) stand for single packets S_3 sent during time intervals $[1, T_1]$, $[T_1+1, T_2]$ and $[T_2+1, T_3]$ respectively.

We have the following propositions for the OPCS-MCF we construct:

Lemma *1:* Given a feasible integer flow F to OPCS-MCF with cost O, we can construct a feasible solution X with objective value $-O$ to OPCS-ILP.

Lemma *2:* Given a feasible solution X to OPCS-ILP with objective value $-O$, we can construct a feasible integer flow F with cost O to OPCS-MCF.

The proof of these two lemmas is omitted due to page limitation, and the theorem follows immediately by Lemmas 1 and 2:

Theorem *3:* Given any optimal integer flow to OPCS-MCF, we can transform it into an optimal solution to OPCS-ILP within polynomial time.

This theorem shows that we can solve the original problem by any minimum cost flow algorithm. Note that the optimal flow found by such algorithms are also integral due to the fact that the capacity of edges in OPCS-MCF are all integers.

V. Numerical Evaluation

In this section, we evaluate the performance of our optimal schedule solution OPT in terms of utility and packet drop rate. We compare OPT with two baseline algorithms, namely maximum weight packet first (MWPF) and urgent packet first (UPF), considering the impacts of flow number and coding opportunity.

A. Evaluation Setup

We focus on the evaluation of the single-hop scenario shown in Fig. 2, the relay node R receives numerous flows and can perform pairwise coding on packets from these flows. R need to make a scheduling policy to sent packets to their corresponding next-hop destination. We perform our simulation under IEEE 802.11 standard which takes about 2 ms to transmit a packet of 160 bytes. This is the same size of VoIP packets using the G.711 codec [19], and we set 2 ms as the length of a time slot in our model. We set the interval to be 40 ms which consists of 20 time slots, and this parameter is also used in [4]. We assume the traffic pattern of video streams to be random because that packets may be retransmitted due to wireless channel error, or experience random delays in queue.

B. Baseline setup

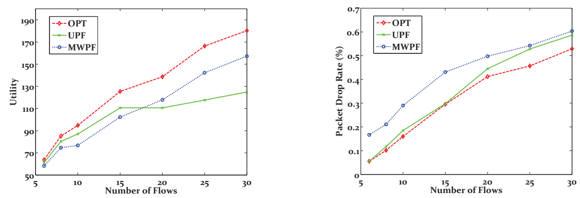
We evaluate the performance of our optimal schedule OPT by comparing it with two baseline algorithms: MWPF and UPF. MWPF preferentially sends the packet with maximum weight among remaining packets, and UPF sends the packet whose delay requirement is most stringent.

C. Performance Comparison

In this subsection, we compare the performance of OPT with MWPF and UPF in terms of utility and packet drop rate, and observe the impacts of flow number and coding opportunity.

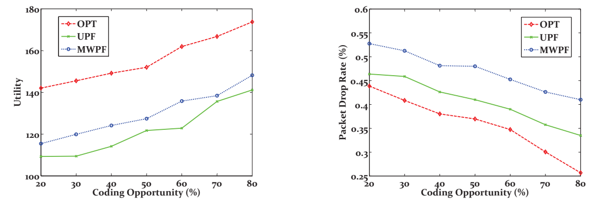
1) Impact of flow number: The loss probability of packet due to channel error is set to be 10% for all downlinks. The number of flows is 6, 8, 10, 15, 20, 25 and 30 respectively. We set the coding opportunity as 60%, i.e. there are 4 single flows and 3 coding pairs among 10 flows. The number of total packets at relay is twice the flow number, i.e. each flow has two packets arrive at the interval on average.

The performance is shown in Fig. 8. When flow traffic is not heavy, there is sufficient time for relay to send packets, so the packet drop rates are low and utilities among three policies are quite close. As the scale of network grows, the superiority of OPT is revealed. Relay can choose packets reasonably by taking into account both deadline constraint and packet weight. UPF may lose the packets with longer delay, and can not make full use of coding opportunities when the traffic is heavy. MWPF only considers the best option instantly without looking further into the queues, which results in a suboptimal solution in the long run. When the traffic is quite beyond the relay's ability to drain, the packet drop rate grows high for all the three policies. Our optimal solution can outperform UPF by up to 65.5% and MWPF by up to 67% in terms of utility,



(a) Average utility under various flow numbers
(b) Average packet drop rate under various flow numbers

Fig. 8. Performance under various flow numbers



(a) Average utility under various coding opportunities
(b) Average packet drop rate under various coding opportunities

Fig. 9. Performance under various coding opportunities

at its packet drop rate is 77.7% that of UPF and 33.3% that of MWPF.

2) Impact of coding opportunities: Fig. 9 evaluates the performance of three policies under different coding opportunities. The number of flows is set to be 20, and we assume there are about 40 packets arrive in a 40 ms long interval. The coding opportunity is set to be 20%, 30%, 40%, 50%, 60%, 70%, 80% respectively. For example, we have 4 coding pairs among 20 flows when the coding opportunity is 20%.

We can see from Fig. 9(a) that OPT increases gradually in utility as coding opportunity grows, and outperforms UPF up to 134.5% and MWPF up to 104.5%. It is obvious that coding opportunity plays a significant role in the improvement of performance of the system. OPT achieves the lowest packet drop rate which is about 76.6% that of the drop rate for UPF and 62% that of MWPF as shown in Fig. 9(b). We can further deduce that MWPF performs better than UPF in terms of utility because it sends packets of larger weights. However this strategy causes higher packet drop rate because it takes no consideration on packet delay constraint.

VI. Conclusion

In this paper, we study the problem of optimal scheduling with flows have pairwise coding opportunities, heterogeneous delay constraints and weights. We formulate the problem as an integer linear programming problem first, and then propose two algorithms to solve the problem. Simulation results show that our proposed solution improves the performance of network, and leads to higher utility and lower packet drop rate.

We are currently extending this idea in several directions, such as considering multihop scenarios, more complicated coding schemes, and more realistic experiments of the wireless network.

Acknowledgment

The work is supported by China The National Science Foundation grants No.61133006, No.61321491, No.61103224, No.61371124, and China 973 projects under Grant No.2014CB340300, No.2012CB316200, and The National Science Foundation of Jiangsu Province under Grant No.BK2011118.

References

[1] V. Vassiliou, P. Antoniou, I. Giannakou, and A. Pitsillides, "Requirements for the transmission of streaming video in mobile wireless networks," in *ICANN*. Springer, 2006, pp. 528–537.

[2] Y. Dai, J. Wu, and C. Xin, "Virtual backbone construction for cognitive radio networks without common control channel," in *INFOCOM*, 2013, pp. 1456–1464.

[3] L. Yang, Y. E. Sagduyu, and J. H. Li, "Adaptive network coding for scheduling real-time traffic with hard deadlines," in *MobiHoc*. ACM, 2012, pp. 105–114.

[4] I. Hou, P. Kumar *et al.*, "Broadcasting delay-constrained traffic over unreliable wireless links with network coding," in *MobiHoc*. ACM, 2011, p. 4.

[5] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," in *Packet Video*. IEEE, 2007, pp. 191–200.

[6] Y.-P. Hsu and A. Sprintson, "Opportunistic network coding: Competitive analysis," in *NetCod*. IEEE, 2012, pp. 191–196.

[7] H. Li, X. Liu, W. He, J. Li, and W. Dou, "End-to-end delay analysis in wireless network coding: A network calculus-based approach," in *ICDCS*. IEEE, 2011, pp. 47–56.

[8] I.-H. Hou and P. Kumar, "Scheduling heterogeneous real-time traffic over fading wireless channels," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[9] I. Hou, R. Singh *et al.*, "Scheduling of access points for multiple live video streams," in *MobiHoc*. ACM, 2013, pp. 267–270.

[10] J. J. Jaramillo, R. Srikant, and L. Ying, "Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 5, pp. 979–987, 2011.

[11] H. Gangammanavar and A. Eryilmaz, "Dynamic coding and rate-control for serving deadline-constrained traffic over fading channels," in *ISIT*. IEEE, 2010, pp. 1788–1792.

[12] C. Zhang, Y. Fang, and X. Zhu, "Throughput-delay tradeoffs in large-scale manets with network coding," in *INFOCOM*. IEEE, 2009, pp. 199–207.

[13] H. Dai, L. Jiang, X. Wu, D. K. Yau, G. Chen, and S. Tang, "Near optimal charging and scheduling scheme for stochastic event capture with rechargeable sensors," in *MASS*. IEEE, 2013, pp. 10–18.

[14] H. Dai, X. Wu, L. Xu, and G. Chen, "Practical scheduling for stochastic event capture in wireless rechargeable sensor networks," in *WCNC*. IEEE, 2013, pp. 986–991.

[15] Z. Ren, P. Cheng, J. Chen, D. K. Yau, and Y. Sun, "Dynamic activation policies for event capture with rechargeable sensors," in *ICDCS*. IEEE, 2012.

[16] X. Li, C.-C. Wang, and X. Lin, "Optimal immediately-decodable inter-session network coding(IDNC) schemes for two unicast sessions with hard deadline constraints," in *Communication, Control, and Computing*. IEEE, 2011, pp. 784–791.

[17] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *SIGCOMM*, vol. 36, no. 4. ACM, 2006, pp. 243–254.

[18] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows: Theory, algorithms, and applications," *Journal of the Operational Research Society*, vol. 45, no. 11, pp. 1340–1340, 1994.

[19] T. Instruments, "Low power advantage of 802.11 a/g vs. 802.11 b," *White paper*, 2003.