PIE: Cooperative Peer-to-Peer Information Exchange in Network Coding Enabled Wireless Networks

Yanfei Fan, Yixin Jiang, Haojin Zhu, *Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—In this paper, we study the issue of scheduling transmission opportunities among nodes (peers) to achieve higher network throughput and lower transmission delay for network coding enabled wireless networks. By conducting an in-depth investigation on the scheduling principles, we propose a cooperative Peer-to-peer Information Exchange (PIE) scheme with an efficient and light-weight scheduling algorithm. PIE can not only fully exploit the broadcast nature of wireless channels, but also take advantage of cooperative peer-to-peer information exchange. Qualitative analysis and extensive simulations demonstrate the effectiveness and efficiency of PIE.**

***Index Terms*—cooperative, peer-to-peer, wireless network coding, scheduling.**

I. Introduction

Network coding has been widely recognized as a promising information dissemination approach to improving network performance [1] by allowing and encouraging coding operations at intermediate network forwarders. Primary applications of network coding include file distribution [2] and multimedia streaming [3] in peer-to-peer (P2P) overlay networks, data persistence in sensor networks [4], and information delivery in wireless networks [5]. Incorporation of network coding into these applications brings many benefits such as throughput improvement [6], energy efficiency [7], and delay minimization [8].

Network coding can be employed to solve the Cooperative Peer-to-peer Repair (CPR) problem [9], where centralized and distributed CPR algorithms are proposed based on observed heuristics. The heuristics reflect some intuitive superficies instead of the essences of network coding based information exchange. In addition, the undetermined parameters in CPR algorithms constitute another open issue: how to tune them to adapt the scheduling algorithms. Such deficiencies motivate us to explore a more insightful scheme to maximize wireless coding gain, i.e., the benefit of combining network coding and wireless broadcast [5].

The scheduling issue in the CPR problem can be reduced to a *peer scheduling* issue by making nodes (peers) send coded packets which are combinations of all packets in a node [2]. Specifically, the *peer scheduling* issue is about how to intelligently schedule the transmission opportunities among peers to maximize the wireless coding gain. Due to the shared wireless channel and the de facto half-duplex transmission feature, peer scheduling policies have a direct impact on the overall network throughput. In many cases, the gap between the optimal and the average is huge.

Most current research focuses on block scheduling problems. Besides opportunistic snooping neighbor states, COPE [5] successfully handles the block scheduling problem by intelligently XOR-ing packets. A multi-partner scheduling scheme [10] employs the Deadline-aware Network Coding technique to adjust the coding window for meeting the time sensitive requirement of media streaming service. An energy-efficient NBgossip scheme [11] utilizes network coding for neighborhood gossip in sensor ad hoc networks. The Rarest First algorithm is advocated through real experiments from being replaced with source or network coding in the Internet [12]. The rarest first idea can be employed in wireless network coding. However, directly applying this idea to peer scheduling is not necessarily optimal.

In this paper, we redefine a *peer scheduling* problem in network coding enabled wireless networks [9]. Based on the summarized peer scheduling principles, we propose a cooperative Peer-to-peer Information Exchange (PIE) scheme with an efficient light-weight peer scheduling algorithm. In addition to the rarest first principle on blocks, we take into consideration the freshness of peers, which is a measurement on how much innovation a peer has against other peers. PIE can not only fully exploit the broadcast nature of wireless channels, but also take advantage of cooperative peer-to-peer information exchange. Qualitative analysis and extensive simulations demonstrate its effectiveness and efficiency.

The remainder of the paper is organized as follows. In Section II, the network model is given. In Section III, we present the peer scheduling principles in network coding enabled wireless networks. PIE is proposed in Section IV. In Section V, the performance of PIE is evaluated in terms of transmission efficiency and computational overhead through extensive simulations, followed by the conclusions in Section VI.

II. Network Model

We consider a network model similar to that in [9]. A remote Base Station (BS) broadcasts a batch of packets (blocks)[1] to nodes. Due to the fading and dynamics of cellular

Manuscript received May 10, 2009; revised October 6, 2009; accepted January 6, 2010. The associate editor coordinating the review of this letter and approving it for publication was J. M. Shea.

Y. Fan and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (e-mail: {yfan, xshen}@bbcr.uwaterloo.ca).

Y. Jiang is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China (e-mail: yixin.tsinghua@gmail.com).

H. Zhu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200030, China (e-mail: zhuhj@cs.sjtu.edu.cn).

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Digital Object Identifier 10.1109/TWC.2010.03.090675

[1]The terms packet and block are used interchangeably in this paper.

channels, each peer receives some (maybe all or none) of these blocks. To mitigate the congestion of downlinks from the BS to those nodes and release the bottleneck of the BS as a network gateway, the nodes can share their received blocks with each other through local wireless networks.

A local wireless network is comprised of several nodes which are also called peers in one-hop wireless scenario. These peers can communicate with each other directly through a commonly shared wireless channel in a half-duplex mode. In other words, if two peers are transmitting at the same time, their signals will interfere with each other, and no peer can correctly receive the signal. On the other hand, due to the broadcast nature of wireless channels, every other peer can receive the signal and recover the frames correctly when one and only one is transmitting.

Without loss of generality, we assume randomly combined packets sent by a peer are linearly independent to each other since the probability of linear dependence is very low [13]. Similarly, coded packets sent out from different peers are also assumed linearly independent to each other.

Table I gives the notations used in this paper.

TABLE I
LIST OF NOTATIONS

Notation	Description
TRN_i	Total Receiving Number of Peer i
DD_i	Deficiency Degree of Peer i
TSN_i	Total Sending Number of Peer i
NUB_i	Number of Unique Blocks of Peer i
$BDM(BDV)$	Block Distribution Matrix i
BRM	Block Rareness Matrix
PDM	Peer Difference Matrix
PFV	Peer Freshness Vector
BAP_j	Benefit of All Peers from the j-th sending operation

III. INVESTIGATIONS ON INFORMATION EXCHANGE PRINCIPLES

Since a specific solution to the peer scheduling problem depends on the original status of the block distribution among the peers, we represent the status as a Block Distribution Matrix (*BDM*). A *BDM* is a (0, 1)-matrix, also known as a binary matrix, in which each element is either one or zero. Row numbers and column numbers of a *BDM* represent peer indexes and block indexes, respectively. In other words, $BDM(i,\ j)$ = 0 means that peer i does not have block j and $BDM(i,\ j)$ = 1 means that peer i has block j. Based on a *BDM*, we summarize the following principles. The correlations between the principles and PIE are discussed in Subsection IV-B.

Definition 1: The total sending number (*TSN*) is defined as the total number of sending operations performed by all peers as a whole for the completion of the information exchange.

Proposition 1: From the viewpoint of peers, a lower bound of *TSN* is the maximum value among all the sums of DD_i and NUB_i, i.e.,

$$TSN \geq \max_i \{DD_i + NUB_i\}, \tag{1}$$

where DD_i is the number of innovative packets that peer i needs to recover the whole original information, and NUB_i denotes the number of the blocks which are uniquely owned by peer i.

Proof: From the viewpoint of peer i, the *TSN* for all peers is equal to the sum of TRN_i and TSN_i, i.e., $TSN = TRN_i + TSN_i$, where TRN_i and TSN_i are the numbers of packets that peer i receives and sends before the completion of information exchange, respectively. Obviously, we have $TRN_i \geq DD_i$ and $TSN_i \geq NUB_i$. Thus, we have $TSN \geq DD_i + NUB_i$. Because the inequality is true for all peers, we have Eq. (1). ■

Proposition 2: From the viewpoint of blocks, a lower bound of *TSN* can be given as follows:

$$TSN \geq \left\lceil \frac{\sum_{i=1}^{N} DD_i}{N-1} \right\rceil, \tag{2}$$

where N is the number of peers ($N \geq 2$).

Proof: For the j-th sending operation, the benefit of all peers (BAP_j) is defined as a cumulative value of the benefits received by all peers. Thus, we have $BAP_j \leq N-1$. On the other hand, each peer has all blocks after the completion of information sharing. Therefore, we have:

$$\sum_{j=1}^{TSN} BAP_j = \sum_{i=1}^{N} DD_i. \tag{3}$$

Thus, we have Eq. (2). ■

Corollary 1: As a summary of Proposition 1 and Proposition 2, a lower bound of *TSN* is:

$$\max\left\{ \left\lceil \frac{\sum_{i=1}^{N} DD_i}{N-1} \right\rceil,\ \max_i \{DD_i + NUB_i\} \right\}. \tag{4}$$

Lemma 1: In the above network model, for any peer i, incoming packets have no innovation to other peers, thus peer i has no necessity to code incoming packets into its future outgoing packets.

Proof: Without loss of generality, let an incoming packet be from peer j. In the above network model, all other peers can receive this packet, which thus has no innovation to those peers any more. In addition, it is peer j that codes this packet, which is a linear combination of all packets peer j has and thus has no innovation to peer j. Therefore, for any peer i, the incoming packet has no innovation to any other peers including peer j and thus peer i has no necessity to include the incoming packet into its future outgoing packets. ■

Proposition 3: In the above network model, sending sequences are order-independent.

Proof: According to Lemma 1, for a given peer sending sequence, switching the orders of any two peers does not change the outcome. In other words, sending sequences are order-independent in the above network model. ■

IV. THE PROPOSED PIE SCHEME

Based on the peer scheduling principles, in this section, we propose a quasi-optimal but efficient and light-weight cooperative Peer-to-peer Information Exchange (PIE) scheme.

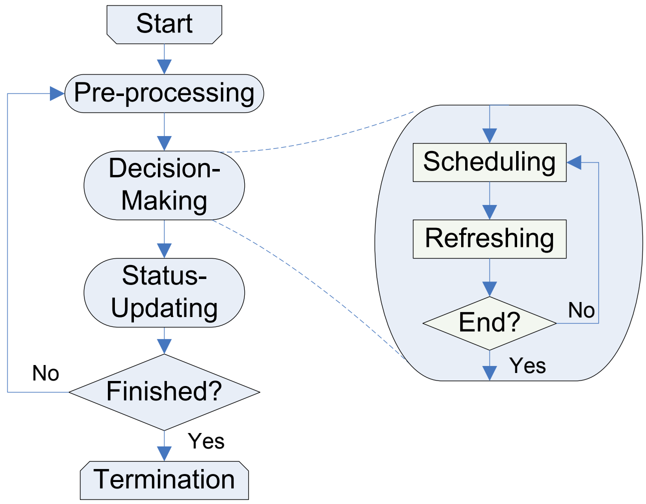

Fig. 1. Flow chart of PIE.

A. The PIE Scheme

The main idea of PIE is to take the freshness of peers into consideration in addition to the rarest first principle on blocks. The basic concept of freshness is a measurement on how much innovation a peer has against all other peers, which can be represented as follows:

$$PFV_i = \sum_j PDM_{ij} = \sum_j \sum_k 1_{\{BDV_{ik} > BDV_{jk}\}}, \qquad (5)$$

where PFV_i denotes the freshness of peer i, PDM_{ij} denotes the difference of peer i against peer j, BDV_i is the block distribution vector of peer i, which is the i-th row vector of block distribution matrix (*BDM*) and so does BDV_j. The indicator function is defined as follows:

$$1_{\{BDV_{ik} > BDV_{jk}\}} = \begin{cases} 1, & \text{if } BDV_{ik} > BDV_{jk}; \\ 0, & \text{Otherwise.} \end{cases} \qquad (6)$$

where BDV_{ik} is the k-th element of the vector BDV_i.

As shown in Fig. 1, PIE consists of four stages: pre-processing, decision-making, status-updating, and termination. The decision-making stage contains two modules with an algorithm in each module. The details of these stages and modules are depicted as follows.

Pre-processing: In PIE, peers first share *BDVs* with each other. The sharing of *BDVs* can be performed by each peer directly broadcasting *BDVs* to others through the shared side channel. Finally, each peer has the block distribution information of all other peers, which forms a *BDM*.

With the *BDM*, each peer can calculate the rareness of blocks and the freshness of peers, which are represented in a Block Rareness Matrix (*BRM*) and a Peer Freshness Vector (*PFV*), respectively. A *BRM* can be calculated as follows. We first calculate the rareness of each block; the rareness of a block denotes the number of peers that have this block; the less the value of the rareness of a block, the rarer the block. The block rareness information is reorganized and put into the *BRM*, where the row number denotes the rareness, the column number denotes the peer number, and the element value denotes the number of blocks of the rareness that a peer has. For example, *BRM*$(i, j) = 3$ means that peer j has 3 blocks of the rareness i. *PFV* is calculated from *PDM*, as defined in Eq. (5). Another data structure is the deficiency degrees (*DD*) of all peers, which is used as the termination condition of the decision-making stage.

Algorithm 1: Peer Scheduling Algorithm

```
Data: BRM, PFV
Result: next_sender
begin
    RBPS ← peers having the rarest blocks in BRM
    MRPS ← peers having the most blocks in RBPS
    if |MRPS| = 1 then
        next_sender ← the unique member of MRPS
    else
        next_sender ← the peer in MRPS with largest freshness
    end
    return next_sender
end
```

Decision-Making: After pre-processing, each peer can start the decision-making stage, which consists of two modules; one is comprised of the peer scheduling algorithm, and the other the status refreshing algorithm.

The peer scheduling algorithm is described in Algorithm 1. First, we choose peers that own the rarest blocks and put them into a peer set with rarest blocks (*RBPS*). Then, peers with most blocks are chosen from the *RBPS* and put into another peer set (*MRPS*). The next sender is the unique peer in *MRPS* if it contains only one member; otherwise, the peer with the largest freshness is chosen as the next sender. The freshness values of peers are taken from *PFV*.

Algorithm 2: Status Refreshing Algorithm

```
Data: BRM, PDM, PFV, DD, next_sender
Result: BRM, PDM, PFV, DD
begin
    v_obj ← a rarest block of the next_sender
    rare ← the rareness of the block v_obj
    APS ← peers having the block v_obj
    foreach peer in APS do
        BRM(rare, peer)−−
    end
    foreach peer in all peers do
        if PDM(next_sender, peer) > 0 then
            foreach member in all peers do
                if PDM(next_sender, member) = 0 then
                    PDM(member, peer)−−
                    PFV(member)−−
                end
            end
            DD(peer)−−
        end
    end
    return BRM, PDM, PFV, DD
end
```

The status refreshing algorithm plays a crucial role in PIE since the refreshed status will affect the next round of peer scheduling. In Algorithm 2, *BRM, PDM, PFV,* and *DD* represent the information of system status from different aspects. *BRM* and *PFV* are for the next round of peer scheduling; *PDM* is for status refreshing; and *DD* is for the termination of the decision-making stage, where the termination condition is that *DD* equals a zero vector.

Notice that many data structures are used instead of a single *BDM*. The reason is that for network coding based information exchange, peers send out coded packets, which make it difficult to keep tracking the status of block distribution information using a single *BDM*. Finally, in the decision-making stage, PIE gives a peer scheduling sequence, which is generated through several rounds of peer scheduling and status refreshing based on the initially shared *BDM*.

Status-Updating: According to the peer scheduling sequence given in the decision-making stage, in this stage, peers send out one coded packet at each time without acknowledgement. Peers keep updating their own block distribution information with the reception of new packets. If a packet is lost, a retransmission from the same peer is required to complete information exchange.

Termination: When each peer recovers all original blocks, the whole process is completed. If those peers have more information for exchange, they can repeat the above process.

B. Discussions

PIE is in line with our summarized principles. For the proof of Proposition 1, we have $TRN_i \geq DD_i$ and $TSN_i \geq NUB_i$. The former principle is observed by PIE, since DD_i is decreased by at most one in each round of scheduling and refreshing in Algorithm 2. The latter is also observed by PIE, since each unique block will make peer i stay in *RBPS*, resulting in that the transmission opportunities will never be scheduled to other peers with only larger-rareness blocks. In other words, from the viewpoint of blocks, before all peers which have unique blocks sends, *DD* will never equal a zero vector since the following equation holds:

$$\sum_{j=1}^{|NUB|} BAP_j = \sum_{j=1}^{|NUB|} (N-1) \leq |DD|, \tag{7}$$

where $|NUB|$ and $|DD|$ are the sums of all NUB_i's and all DD_i's, respectively. Thus, PIE is naturally in accordance with the Proposition 1. Moreover, according to Algorithm 2, we can see that the BAP_j is no larger than $N-1$, making PIE conform to the Proposition 2. Finally, following Proposition 1 and Proposition 2, the Corollary 1 naturally holds.

From Eq. (5), it can be seen that freshness is a cumulative difference of a peer against other peers. Thus, the concept of freshness represents a measurement of possible innovation a peer has against other peers. This definition captures the essence of network coding based information exchange in terms of innovative information, thus assisting to maximize the wireless coding gain.

V. Performance Evaluation

To verify the effectiveness and efficiency of PIE, we conduct extensive simulations for performance evaluation. In our simulation, each peer can successfully receive the original blocks from a BS with a prescribed probability. We define the probability as the sparsity degree of the original blocks. The performance of PIE is evaluated and also compared with the rarest first algorithm in terms of transmission efficiency and computational overhead.

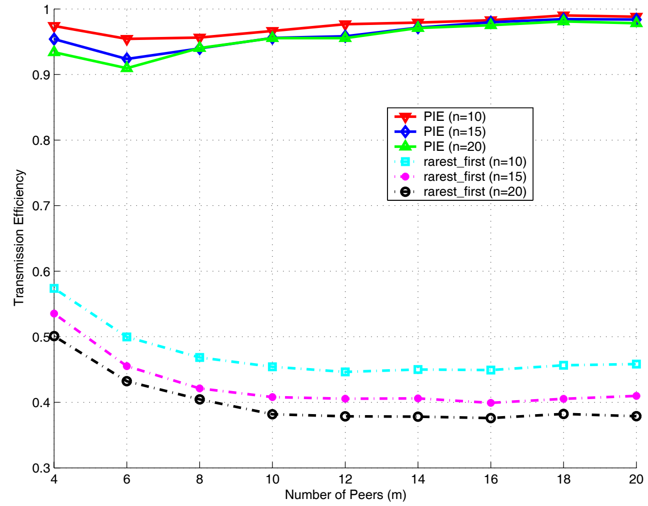

Fig. 2. Transmission efficiency vs. number of peers (sparsity = 0.8, n: number of blocks).

The theoretical lower bound of *TSN* in Corollary 1 is adopted as the benchmark for the evaluation of transmission efficiency, which is defined as:

$$E_t = \frac{TTA/TSN}{TTA/LB} = \frac{LB}{TSN}, \tag{8}$$

where E_t is transmission efficiency, *TTA* is the total transmission amount of information exchange, *TSN* is our simulation result, and *LB* is the theoretical lower bound of *TSN*, as defined in Eq. (4). From the definition, we know $E_t = LB/TSN \leq OPT/TSN \leq 1$ since $LB \leq OPT \leq TSN$, where *OPT* denotes the optimal solution. Thus, we know PIE is near to the optimal in terms of transmission efficiency when E_t is near to 1.

A. Transmission Efficiency

Fig. 2 shows the transmission efficiency versus the number of peers. It can be seen that the transmission efficiency of PIE is much higher, about 30% on average, than that of the rarest first algorithm. With the increase of the number of peers, the transmission efficiency of PIE increases and almost reaches its theoretical upper bound. Simulation results with different numbers of blocks (n = 10, 15, 20) are given for extensive verification.

The transmission efficiency versus the number of blocks is shown in Fig. 3. It can be seen again that PIE outperforms the rarest first algorithm. With the increase of the number of blocks, the transmission efficiencies of both schemes decrease; while PIE still maintains more than 95% transmission efficiency in almost all scenarios. Simulation results with different numbers of peers (m = 4, 8, and 12) are shown respectively for extensive verification. A more extensive comparison between PIE and the rarest first algorithm in terms of transmission efficiency is shown in Fig. 4.

Fig. 5 shows the transmission efficiency versus the sparsity degree with different numbers of peers (m = 5, 10, and 15) and different numbers of blocks (n = 5, 10, and 15). Both schemes have the almost same changing trend, while PIE outperforms the rarest first algorithm with extensively diverse sparsity degrees.

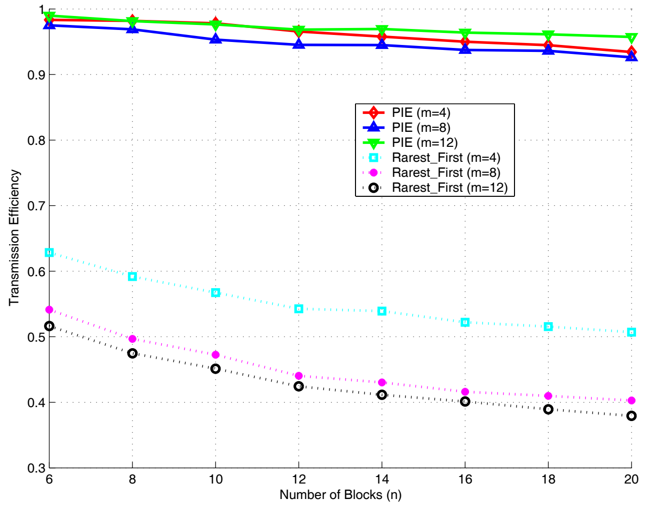

Fig. 3. Transmission efficiency vs. number of blocks (sparsity = 0.8, m: number of peers).

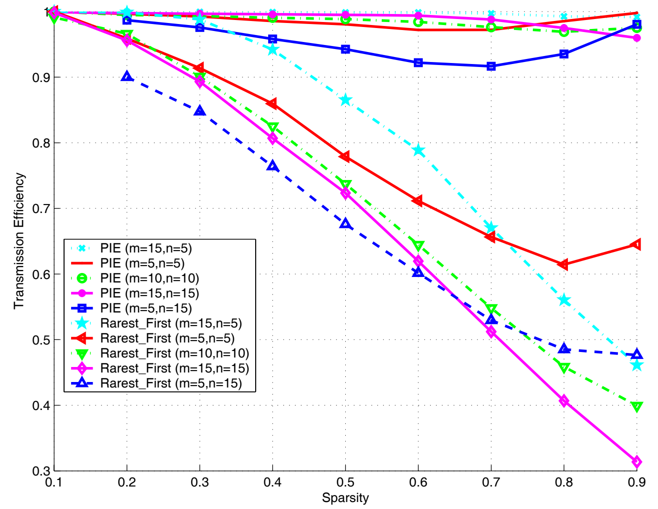

Fig. 5. Transmission efficiency vs. sparsity (m: number of peers, n: number of blocks).

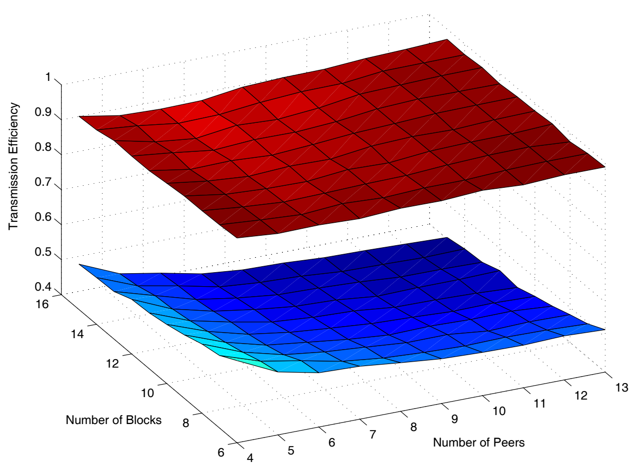

Fig. 4. PIE vs. rarest first (sparsity = 0.8).

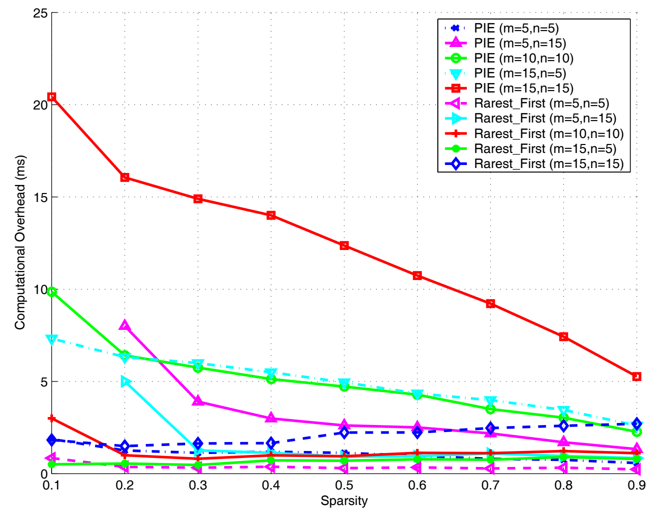

Fig. 6. Computational overhead vs. sparsity (m: number of peers, n: number of blocks).

B. Computational Overhead

The computational overheads of PIE and the rarest first algorithm are shown in Fig. 6, where all the computational overheads are collected from a laptop platform with a CPU of Intel Pentium M 1.8GHz and a RAM of 512MB. It can be seen that the computational overhead of the rarest first algorithm increases almost linearly with the sparsity degree, while that of PIE decreases almost linearly. Furthermore, the computational overhead of PIE still remains in the range of practical applications.

VI. Conclusions

In this paper, we have proposed a cooperative Peer-to-peer Information Exchange (PIE) scheme with a compact, efficient, and light-weight peer scheduling algorithm for network coding enabled wireless networks. PIE can not only fully exploit the broadcast nature of wireless channels, but also take advantage of cooperative peer-to-peer information exchange. Qualitative analysis and extensive simulations have demonstrated the effectiveness and efficiency of PIE. Our future work will focus on designing efficient peer scheduling schemes for multi-hop wireless networks.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204-1216, July 2000.

[2] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE INFOCOM'05*, vol. 4, pp. 2235-2245, Mar. 2005.

[3] M. Wang and B. Li, "Network coding in live peer-to-peer streaming," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1554-1567, 2007.

[4] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: maximizing sensor network data persistence," in *Proc. ACM SIGCOMM'06*, pp. 255-266, 2006.

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

[6] P. A. Chou and Y. Wu, "Network coding for the Internet and wireless networks," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 77-85, Sept. 2007.

[7] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Trans. Networking*, vol. 16, no. 2, pp. 450-463, Apr. 2008.

[8] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, “Wireless broadcast using network coding," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 914-925, Feb. 2009.

[9] X. Liu, S. Raza, C.-N. Chuah, and G. Cheung, “Network coding based cooperative peer-to-peer repair in wireless ad-hoc networks," in *Proc. IEEE ICC’08*, pp. 2153-2158, May 2008.

[10] H. Chi, Q. Zhang, J. Jia, and X. Shen, “Efficient search and scheduling in P2P-based media-on-demand streaming service," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 1, pp. 119-130, Jan. 2007.

[11] F. Lu, L.-T. Chia, and K. L. Tay, “NBgossip - neighborhood gossip with network coding based message aggregation," in *Proc. IEEE MASS’07*, pp. 1-12, Oct. 2007.

[12] A. Legout, G. Urvoy-Keller, and P. Michiardi, “Rarest first and choke algorithms are enough," in *Proc. 6th ACM SIGCOMM Conf. Internet Measurement (IMC’06)*, pp. 203-216, 2006.

[13] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, “An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *Proc. IEEE INFOCOM’09*, Rio de Janeiro, Brazil, Apr. 2009.