

Research Article

Efficient Coding-Based Scheduling for Multi-Item Requests in Real-Time On-Demand Data Dissemination

Shujuan Wang, Chunting Yan, and Zhengtao Yu

School of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan 650500, China

Correspondence should be addressed to Shujuan Wang; shujuanwang0703@126.com

Received 23 February 2016; Accepted 8 June 2016

Academic Editor: Juan C. Cano

Copyright © 2016 Shujuan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

On-demand broadcast is an effective data dissemination technique to enhance the system ability to handle dynamic client access patterns. With the development of real-time wireless applications, it is practical for clients to request multiple related data items with time constraint in one request. For real-time data broadcast, items are only useful if they can be received by clients before expiration; otherwise, the related requests become invalid. Hence, minimizing the number of requests that miss their deadlines becomes an important task in on-demand broadcast, especially for multi-item scenario, where more requests are inclined to miss their deadlines due to the heavier workload and greater complexity, compared with single-item scenario. In this paper, we propose a network coding-based scheduling algorithm for real-time multi-item data dissemination. The proposed algorithm exploits the coding opportunities between cached and requested data items and integrates network coding with data scheduling to minimize the deadline miss ratio. Extensive simulation results demonstrate that the proposed algorithm achieves the lowest deadline miss ratio under various circumstances. In addition, the high efficiency of the proposed algorithm in utilizing the transmission bandwidth is proved for that fewer instances of ineffective service are provided for satisfying clients' requests within time constraints.

1. Introduction

With the rapid development of communication technologies, modern mobile communication devices have obtained the capabilities of transmitting, storing, and processing information independently, enabling the rush of new applications with the purpose of safety, being economic, and infotainment [1–4]. Meanwhile, data acquisition becomes increasingly important for these emerging mobile applications [5–8].

Data broadcast is proven to be a highly efficient approach for disseminating information due to its potential of satisfying all outstanding requests for the same data item with a single response [9, 10]. Two types of data dissemination are studied extensively in literatures. One is the push-based data broadcast and the other is the pull-based data broadcast, namely, on-demand data dissemination. The push-based data broadcast periodically broadcasts data items according to a static schedule which is computed offline from clients' historical data access statistics and is efficient with applications that require a small set of data items with stable access pattern [11–13]. On the contrary, on-demand data dissemination

schedules requests in the service queue and broadcasts data based on various attributes of pending data items at the server. By considering different clients' requirements for data acquisition, on-demand broadcast is more widely used for dynamic, large-scale data dissemination and attracts more and more research interests in recent years [14–17]. Meanwhile, data items are related and have temporality for many popular information services, such as the traffic condition inquiries, the weather report queries, and the stock quotes. Data may become invalid or less useful as time passes. Therefore, the clients usually request the data items with deadline constraints and the server should broadcast the requested data items before their deadlines to ensure that the data transmitted to the clients are useful and informative. In this paper, we focus on the on-demand multi-item requests data broadcast with deadline constraints. The main performance objective is the deadline miss ratio, which is defined as the percentage of requests that miss their deadlines.

Although various scheduling algorithms for on-demand broadcast have been proposed in the literature [18–22], these algorithms do not fully utilize the strength of data broadcast

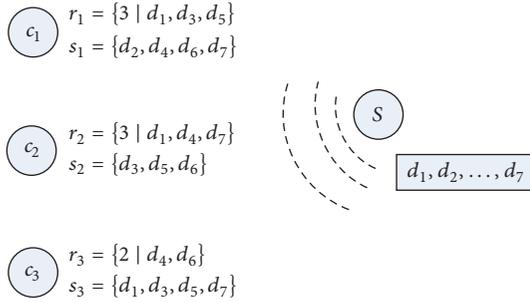


FIGURE 1: An example of on-demand multi-item data dissemination with time constraint.

and only clients requesting the same data item can be satisfied in one broadcast slot. With network coding technique [23], it is possible that clients requesting different data items can be satisfied simultaneously by utilizing previously received data items stored at clients. Thus, the access time and the deadline miss ratio can be further reduced.

Consider an on-demand multi-item broadcast with time constraint in Figure 1, which consists of one server S and three clients c_1 , c_2 , and c_3 . Assume that the server has seven data items, d_1, d_2, \dots, d_7 . Client c_1 has already received and stored four data items from previous broadcast slots, denoted as $s_1 = \{d_2, d_4, d_6, d_7\}$, and still needs items $r_1 = \{3 \mid d_1, d_3, d_5\}$ within three broadcast slots; the number in r_1 denotes the deadline for c_1 's request r_1 . Similarly, client c_2 has received $s_2 = \{d_3, d_5, d_6\}$ and requests for $r_2 = \{3 \mid d_1, d_4, d_7\}$ in three slots; c_3 has cached $s_3 = \{d_1, d_3, d_5, d_7\}$ and needs to receive $r_3 = \{2 \mid d_4, d_6\}$ in two broadcast slots. Further assume that the size of each data item is 1 unit and the broadcast bandwidth is $B = 1$ unit/slot; the server can broadcast one data item in each time slot. With the traditional noncoding scheduling strategy, which aims at maximizing the number of clients a data item can satisfy in each broadcast slot, the server will broadcast d_1 , d_4 , and d_7 sequentially. Unfortunately, client c_3 's request r_3 missed its deadline without receiving $\{d_6\}$ after two broadcast slots and client c_1 's request r_1 missed its deadline without receiving $\{d_3, d_5\}$ in three slots. Finally, only client c_2 received the requested data items on time and thus the deadline miss ratio can be calculated as $2/3 = 67\%$. Contrarily, with the support of network coding, the server finds out an optimal scheduling decision sequence so as to broadcast $d_3 \oplus d_4$, $d_1 \oplus d_6$, and $d_5 \oplus d_7$ orderly. By receiving $d_3 \oplus d_4$, c_1 can get d_3 since it has d_4 ; c_2 and c_3 can both get d_4 by using cached d_3 to decode the coded packet. Similarly, c_1 and c_2 decoded d_1 and c_3 decoded d_6 from the second broadcast $d_1 \oplus d_6$. After that, client c_3 has received all data items it requested on time and gets satisfied. Finally, the third broadcast of coded packet $d_5 \oplus d_7$ provided d_5 for c_1 and d_7 for c_2 , and both clients get satisfied within their deadlines. In this way, no client misses a single deadline and the deadline miss ratio of the network coding broadcast strategy is 0. As illustrated in this example, by utilizing the information about clients' cached data items, network coding-based broadcast can improve the dissemination performance evidently.

The impact of network coding on the deadline miss ratio of real-time on-demand data dissemination is determined by the encoding decision of the server at each broadcast slot. However, it is difficult to model the encoding decision process since the clients' requests are generated randomly, and the multi-item request characteristic further complicates the problem. Thus, it is highly desirable to design an efficient data broadcast strategy for on-demand multi-item data dissemination with deadline constraint. This paper proposes a data scheduling algorithm which incorporates network coding technique into the real-time multi-item data broadcasting. Specifically, the proposed algorithm considers not only broadcasting each single data item as soon as possible, but also fulfilling multi-item request on time as much as possible, in order to minimize the total deadline miss ratio and to improve the broadcast efficiency. The main contribution of this paper can be summarized as follows:

- (i) We use the network coding technique to improve the PVC algorithm [24] for real-time multi-item data broadcast. The PVC algorithm designs the scheduling weight by imitating the economic's measure of profitability and achieves good performance compared with traditional scheduling algorithms. However, the scheduling performance can be further improved by utilizing network coding to encode the most weighted data item with multiple less weighted data items together to broadcast at the same broadcast slot, as long as the decoding condition is met. We demonstrate that network coding can affect the broadcast performance positively.
- (ii) We use the maximal weight clique search [25–27] to solve the encoding decision problem for real-time multi-item data broadcast. The weight of each vertex is calculated according to the novel definitions as in PVC algorithm. We present how to integrate the network coding technique with the maximal clique search without adding too much overhead.
- (iii) The simulation results show that the proposed network coding-based algorithm achieves better performance under various scenarios as compared with a number of representative scheduling algorithms.

The remainder of this paper is organized as follows. Section 2 presents the background of the research area. Section 3 describes the system architecture. Section 4 outlines the proposed algorithm and explains the coding strategy in detail. Section 5 evaluates the proposed algorithm's performance through extensive simulations. Finally, Section 6 concludes the paper.

2. Related Works

2.1. Data Scheduling in On-Demand Data Dissemination. On-demand data broadcast is proven to be an efficient way to disseminate information to a large population of mobile clients. In previous research works, data scheduling algorithms play a very important role in reducing system response time and satisfying clients efficiently. Various kinds

of scheduling algorithms have been proposed to decide the broadcast sequence of requested data items pending in the service queue of the server.

Dykeman et al. proposed three scheduling algorithms, FCFS (First-Come-First-Serve), MRF (Most-Request-First), and LWF (Longest-Wait-First) [18]. FCFS broadcasts the requested data items according to their arrival sequence. MRF schedules broadcasting the data item with the largest number of pending requests. In the LWF scheme, the data item which has the largest waiting time, equal to the sum of the time that all requests for this data item have been waiting, is broadcasted first. Aksoy and Franklin proposed the $R \times W$ algorithm which considers the number of pending requests for the data item and the time that the request has been waiting together [19]. $R \times W$ schedules the data item with the maximal value of $R \times W$ first, where R denotes the number of requests referring to the data item and W is the time that the oldest request for the data item has been waiting.

The constantly changing characteristics of data item size in on-demand data broadcast are studied in [20]. The authors proposed two scheduling models under different split strategies, one is the equal split scheduling model $ES-L \times R \times W$ which proposes the equal splitting strategy (ES) and a deadline adjust strategy. The other is the unequal split scheduling model $US-L \times R \times W$ which proposes the unequal split strategy (US) and two effective scheduling algorithms, namely, priority first (PF) and priority and bandwidth first ($P \times BF$). Both scheduling models and the split strategies are proven to be able to improve bandwidth utilization and dynamically adjust to variation of data item size. A new metric called stretch is introduced to evaluate the performance with different data sizes in [28]. Stretch is the ratio of the access time of a request to its service time. A scheduling algorithm LTSF (Longest Total Stretch First) based on the concept of stretch is proposed that the data item with the largest total current stretch is broadcasted first. In [29], Sharaf and Chrysanthis proposed the STOBS which computes the value $(R \times W)/S$ of each requested data item in the queue and schedules the item with the highest $(R \times W)/S$, where R , W , and S are the number of pending requests, the waiting time of the first request, and the size of the data item, respectively. All these algorithms consider the attribute of data size.

Xuan et al. [21] examined several broadcast transmission scheduling policies for timely on-demand broadcast and indicate that EDF (Earliest Deadline First) scheduling algorithm achieves good performance. After that, Xu et al. [22] proposed the so-called $SIN-\alpha$ (Slack time Inverse Number of pending requests) scheduling algorithm for time-critical on-demand broadcast, which considers the urgency and number of outstanding requests together and broadcasts the data item with the minimum value of $\sin-\alpha$, where $\sin-\alpha = \text{slack}/\text{num}^\alpha$, slack is the duration from current time to the deadline of the most urgent request for the data item, and num is the number of pending requests for the data item. All the above works only considered the on-demand data broadcast in single-item request scenario, where mobile clients can submit a request for only one data item from each broadcast unit, whereas many on-demand broadcast applications require multiple

data items in one request and thus cannot be run by these algorithms effectively.

In [24], Lv et al. proved the NP-hardness of broadcast scheduling of real-time multi-item requests in single channel environment and proposed a profit-based scheduling algorithm named PVC, which utilizes the novel concept “profit” of pending items and “opportunity cost” of pending requests. Simulation results showed distinct improvement of PVC in comparison with EDF and $SIN-\alpha$ in real-time multi-item requests environment. However, this algorithm schedules only one data item for each broadcast, which makes the insufficient utilization of bandwidth and undermines the performance.

2.2. Network Coding-Based Data Broadcast. Network coding (NC) was firstly proposed by Ahlswede et al. in 2000 [23]. It has been proved that the theoretical upper bound of wireless multicast transmission rate, that is, the max. flow of a network, can be achieved by employing network coding in a wireless network [30, 31]. Meanwhile, due to the broadcast nature of wireless channels, wireless networks exhibit significant data redundancy that the transmitter delivers the same packet to multiple nodes within its radio range at each hop. Network coding can be exploited to reduce the redundancy, increase the overall network throughput, and decrease the transmission delay [32–34]. NC has shown great potential to improve the bandwidth efficiency of wireless broadcast systems. The design of NC-based on-demand data broadcast schemes has received significant research interests and many network coding-based data scheduling algorithms have been proposed for on-demand single-item requests environment [35–40]. Hou studied a model that jointly considers the application requirements on traffic patterns, delay constraints, and throughput requirements, as well as wireless limitations, including the unreliable wireless links and the lack of feedback information. Based on this model, a general framework was developed for designing feasibility-optimal broadcasting policies that apply to systems with various network coding mechanisms [35]. In [36], a coding assisted scheduling algorithm for on-demand data broadcast named ADC-1 and its enhanced version ADC-2 are proposed for single-item requests environment. In [37], Zhan et al. propose a generalized encoding framework for incorporating network coding into data scheduling algorithms for on-demand broadcast. Data scheduling is formulated as a clique searching problem in a graph and network coding is migrated into the scheduling algorithms to optimize the performance with consideration of different attributes. Performance evaluation of the proposed encoding framework is given by [38]. Wang and Liang considered the adoption of the traditional linear network coding in on-demand data broadcast and proposed a network coding algorithm which utilizes the correlation which exists between received and requested data items to reduce the completion delay [39]. The fairness between different receivers is well considered. In [40], Wang et al. designed a joint rate selection and wireless network coding (RSNC) scheme with delay constraint so as to maximize the total benefit of the packets that are successfully received at the destinations without missing their deadlines. However, in

At the beginning of each broadcast slot

- (1) **for** Each arrived multi-item request Q **do**
- (2) Insert Q into the requests' queue
- (3) **end for**
- (4) **while** The requests' queue is not empty **do**
- (5) **for** Each multi-item request Q **do**
- (6) Calculate Q 's scheduling weight according to Definitions 1–10, insert Q into the service queue in descending order of scheduling weight $PVC(Q)$;
- (7) **end for**
- (8) Select the request with the most scheduling weight as the schedulable request Q^* , further select the data item d_j with the most absolute profit, $d_j \in Q^*$, as the seed for network coding. That is $seed \leftarrow d_j = \arg \max_{d_i \in Q^*} \text{profit}(d_i)$
- (9) Construct the instantly decodable coding graph G as described in Section 4.2.2
- (10) Find the maximal clique in graph G which contains the seed item d_j and XOR all items in this clique as the encoded packet, and broadcast the packet to all clients.
- (11) Update each client's status of received and request data items. Renew the requests' queue and the service queue by removing requests which get satisfied or miss their deadlines.
- (12) **end while**

ALGORITHM 1: NC-based PVC scheduling algorithm.

these works, mobile clients can only ask for one data item in each request, which can not represent the requirements of emerging wireless applications and cannot fully utilize the advantage of network coding.

In contrast, we propose a NC-based scheduling algorithm (Algorithm 1) for on-demand multi-item requests data broadcast with time constraints, which combines the performance advantage of scheduling algorithm and network coding. The motivation is to provide efficient broadcast for satisfying multi-item requests within their deadlines. Using network coding technique, we can improve the bandwidth utilization and increase the network throughput, thus minimizing the number of requests that miss their deadlines.

3. System Model

We consider the general system model of on-demand data broadcast, which consists of one server and a number of clients. In Figure 2, each client has a local cache to store the requested data items received from the server. The cache space is limited and certain cache replacement policy is applied [37]. When a client cannot find data items in its cache, the client sends a request for the data items and its updated cache content to the server through the uplink channel. After sending the request, the client listens to the broadcast channel and retrieves its requested data items through the server's broadcasts. Each request is associated with a deadline beyond which the request becomes invalid. A data item can be considered as a fixed-size page in the database and the request refers to multiple data items. Given bandwidth $B = 1$, the time taken to broadcast a data item is called a broadcast tick (time slot).

Upon receiving a request, the server inserts it into the service queue. A request waits to be scheduled in the service queue until its requested data items are broadcasted. Firstly, the server retrieves the requested data items stored in the local database by applying certain scheduling algorithms; it then encodes the selected data items based on information

about clients' cached and requested data items. Lastly, the server broadcasts the encoded packet through the downlink channel. A client gets satisfied if its request has been served successfully, which means the client has received all the data items correctly corresponding to its request before deadline expiration. A request will be removed from the service queue once it exceeds its time limit. In this model, we apply the simple XOR encoding technique while encoding/decoding data items, merely due to the easy implementation and less overhead of XOR [41–43].

4. The Proposed NC-Based Scheduling Algorithm

4.1. Motivation. Many of the recent on-demand data broadcast research works have shown that network coding can help clients retrieve different data items simultaneously from the server. However, to the best of our knowledge, most of them only consider the single-item request scenario and no existing solution that effectively integrates data scheduling and network coding for on-demand multi-item data broadcast exists. In this paper, we propose an algorithm to exploit the advantage of combining network coding and data scheduling. The primary goal of the scheduling algorithm is to minimize the request deadline miss ratio and equally to maximize the number of requests which meet their deadlines. The proposed algorithm carries forward the fine performance of the data scheduling and covers the shortage of it through the integration with network coding.

4.2. The Proposed Algorithm

4.2.1. Preliminaries. Through the paper, we assume that the time is slotted. We further assume that the length of each data item stored at the server is fixed as 1 unit length, and the bandwidth between a client and the server is $B = 1$. Given these assumptions, at each time slot, the server can transmit

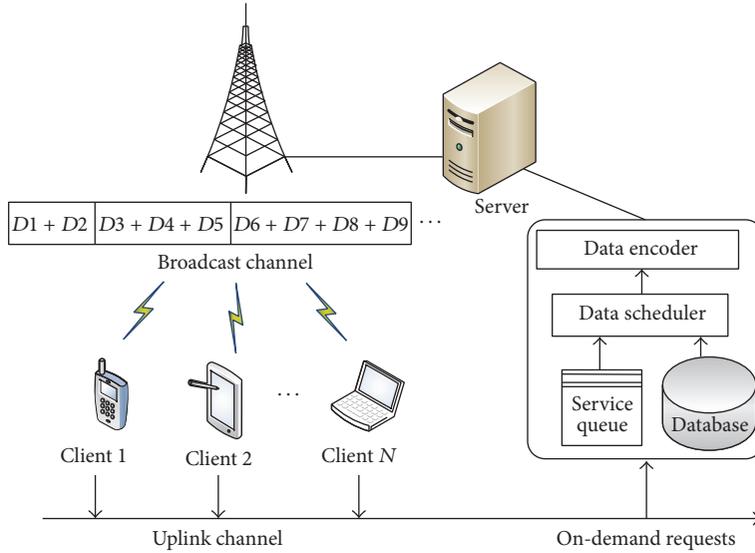


FIGURE 2: System model.

one data item or a coded packet with the length of one unit. The proposed algorithm is inspired by the fine work of Lv et al. [24]. In [24], the authors proposed a new concept “profit” to characterize the potential contribution of a pending item to system performance and give definitions of some related concepts as follows (for more comprehensive explanation of the definitions, please refer to [24]).

Definition 1 (slack time). The remaining time slots before the deadline of a request are denoted as s . A request is a live request only when it can meet its deadline and the slack time is enough for broadcasting all its unserved data items.

Definition 2 (live request). Consider a request $Q(s, k)$ which consists of k unserved (not received by a client which requests it) data items with slack time s . $Q(s, k)$ is said to be live if $s \geq k \geq 0$ (since $B = 1$, the time required to transmit k unserved data items equals to k time slots. k denotes the number of time slots needed to transmit the total unserved data items). Otherwise, the request is dead. When $k = 0$, the request is completed (satisfied) as it has no unserved data items.

Definition 3 (relative profit). Concerning a request Q for n items with an unserved item set of cardinality k , the relative profit of an item d_i is defined as

$$\text{profit}(d_i | Q) = \begin{cases} \frac{n-k+1}{n}, & \text{if } d_i \in Q \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Since $(n - k)$ is the number of data items which have been served during the past scheduling process, the relative profit actually denotes the contribution for request Q via broadcasting item d_i . Meanwhile, noticing that a data item can serve all requests pending for it simultaneously, an absolute profit for the item can be defined as follows.

Definition 4 (absolute profit). The sum of relative profits of an item d_i with respect to all pending requests for d_i in the service queue is as follows:

$$\text{profit}(d_i) = \sum_{j=1}^m \text{profit}(d_i | Q_j). \quad (2)$$

Absolute profit reflects the contribution of broadcasting item d_i for the whole system performance.

Definition 5 (average profit). Given a request Q with an unserved items set of cardinality k , the average profit for serving Q is defined as

$$\text{avg profit}(Q) = \frac{\sum_{d_i \in Q} \text{profit}(d_i)}{k}. \quad (3)$$

The average profit measures the contribution of serving a request to system performance. A request has a high average profit when its unserved data items have high absolute profit. A data item is profitable if it is popular (hot) and its broadcast can satisfy more requests or bring more requests closer to completion.

Definition 6 (relative compatible or exclusive requests). Request $Q_j(s_j, k_j)$ is exclusive to request $Q(s, k)$ if $k + k_j - |Q \cap Q_j| > s_j$; otherwise Q_j is compatible with Q . By the definition, if Q_j is exclusive to Q , Q_j will be lost if Q is selected to broadcast until satisfaction. Note that if Q_j is exclusive to Q , Q is not necessarily exclusive to Q_j .

Definition 7 (request expected revenue). Given a request $Q_j(s_j, k_j)$, its expected revenue with respect to request $Q(s, k)$ is defined as

$$E(Q_j | Q) = \begin{cases} \frac{k}{k_j}, & \text{if } Q_j \text{ is exclusive to } Q \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The request expected revenue is a metric value to quantify other alternatives with respect to a scheduling decision and can be interpreted as the gain of the scheduling decision of serving request Q which would be abandoned in favor of Q_j .

Definition 8 (item expected revenue). Given an item d_i , its expected revenue with respect to request $Q(s, k)$ is the sum of expected revenues of all requests for d_i with respect to Q :

$$E(d_i | Q) = \sum_{Q_j: d_i \in Q_j} E(Q_j | Q). \quad (5)$$

Definition 9 (opportunity cost). Given a request $Q(s, k)$, the opportunity cost is defined as the maximum expected revenue with respect to Q for all items *NOT* requested by Q :

$$C(Q) = \max_{d_i \notin Q} \{E(d_i | Q)\}. \quad (6)$$

The opportunity cost represents the next best alternative scheduling data item's expected revenue, which is abandoned in favor of scheduling Q . Of course, only items in the requests exclusive to Q can generate nonzero revenue.

Definition 10 (scheduling weight). The scheduling weight of a request $Q(s, k)$ is defined as follows:

$$\text{PVC}(Q) = \frac{\text{avg profit}(Q) - C(Q)}{s(Q)}, \quad (7)$$

where $s(Q)$ is the slack time of request Q .

This PVC weight is used in the item level scheduling algorithm named Profit Versus Cost (PVC). The basic idea of PVC is to first choose the request with the largest scheduling weight and then broadcast the item with the largest absolute profit in the selected request.

4.2.2. The Instantly Decodable Network Coding Graph.

Assume there are N clients in the system, denoted as $U = \{u_1, u_2, \dots, u_N\}$, and a server S whose database D contains M data items, where $D = \{d_1, d_2, \dots, d_M\}$. Each client stores data items that were broadcasted by the server and generates a stream of multi-item requests containing data items that the client cannot find in its cache. To integrate network coding into on-demand data dissemination, the server needs to know the information about requests submitted by multiple clients, as well as the cached data items at each client to make proper scheduling decisions. Assuming that clients can piggyback their caches' status in their requests sent to the server, we can construct the graph model to operate the encoding decision.

Let W_i be the set of data items that are requested by client c_i and let H_i be the set of data items that are stored in the cache of client c_i ; a graph $G(V, E)$ can be constructed by first generating a vertex $v_{ij} \in V$ for each data item $d_j \in W_i$, $1 \leq i \leq N$. Two vertices v_{ij} and v_{kl} are connected by an edge in E if one of the following condition is true:

- (i) $j = l$: the two vertices are induced by the loss of the same item d_j by two different receivers c_i and c_k .

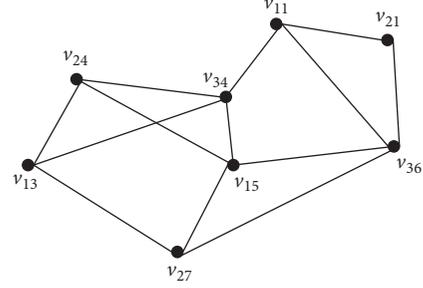


FIGURE 3: The instantly decodable network coding graph constructed from the example in Figure 1.

- (ii) $d_j \in H_k$ and $d_l \in H_i$: the requested item d_j is in the Has set of client c_k and vice versa. In other words, client c_k has cached item d_j and client c_i has cached item d_l , while d_j is requested by c_i and d_l is requested by c_k .

From the definition of G , the instantly decodable network coding graph corresponding to the example in Figure 1 is constructed as in Figure 3. In Figure 3, vertices v_{11} , v_{13} , and v_{15} represent the multi-item request from client c_1 for data items d_1 , d_3 , and d_5 . Similarly, vertices v_{21} , v_{24} , and v_{27} represent the multi-item request from client c_2 for data items d_1 , d_4 , and d_7 , and vertices v_{34} and v_{36} represent the multi-item request from client c_3 for data items d_4 and d_6 .

Since clients c_1 and c_2 request the same data item d_1 , there is an edge (v_{11}, v_{21}) between vertices v_{11} and v_{21} . Also, as client c_2 and c_3 request the same data item d_4 , there is an edge (v_{24}, v_{34}) between vertices v_{24} and v_{34} . Furthermore, according to the above second condition, there exists an edge between vertices v_{11} and v_{34} since $d_1 \in H_3$ and $d_4 \in H_1$, which means that c_3 has d_1 requested by c_1 , and c_1 has d_4 requested by c_3 . Therefore, the two requested data items can be received in one broadcast slot by sending $d_1 \oplus d_4$. The rest of the edges are connected in a similar way.

In the graph model, each vertex represents a data item requested by a client. Since we consider the multi-item data dissemination scenario, there will be multiple vertices associated with the same request in the graph. As mentioned, client c_1 submitted a request for d_1 , d_3 , and d_5 , and there exist vertices v_{11} , v_{13} , and v_{15} . However, any two of the three vertices are not adjacent, because neither of the above two conditions could be true. In other words, a client cannot submit two different requests for the same data items and cannot submit a request for a data item which is already available in its own cache.

A clique in an undirected graph is defined as a subset of vertices such that every two vertices in the subset are adjacent. According to [25, 27, 37], let $C = \{v_{i_1 j_1}, v_{i_2 j_2}, \dots, v_{i_k j_k}\}$ be a clique in G , $D_C = \{d_j \mid v_{ij} \in C\}$ denotes the data items represented by the vertices in C , and $R_C = \{c_i \mid v_{ij} \in C\}$ denotes the involved clients set. Note that some of the data items may be repeated in D_C (same items requested by different clients); further denote D_C as $D_C = \{d_{c_1}, d_{c_2}, \dots, d_{c_t}\}$ with $t \leq k$. For all $c_i \in R_C$ and $v_{ij} \in C$, c_i can obtain d_j after receiving items

combination $d' = d_{c_1} \oplus d_{c_2} \cdots \oplus d_{c_k}$. It can be easily inferred that any packet, generated by encoding the source items which are identified by the vertices of any clique in G , is an instantly decodable packet, which means that the encoded packet can be decoded by all clients immediately upon successful receiving. Furthermore, a maximal clique is a clique that cannot be extended by including one more adjacent vertex, that is, a clique such that there is no clique with more vertices.

4.2.3. The NC-Based PVC Algorithm. The PVC scheduling weight is designed by imitating the economics' measure of profitability where net profit is equal to the profit minus cost. As defined above, the average profit can faithfully describe the profit of each request in the service queue and thus little room for improvement exists. However, the performance gain can be improved by reducing the cost $C(Q)$. Given the average profit $\text{avg profit}(Q)$ and the slack time $s(Q)$, the scheduling weight $\text{PVC}(Q)$ increases with the reduction of the opportunity cost $C(Q)$. Noticing that the PVC scheduling weight $\text{PVC}(Q)$ is actually a measure of performance gain introduced by scheduling request Q , with less (or none, if possible) opportunity cost, the algorithm receives more profit and achieves better performance.

To minimize the opportunity cost, network coding technology is introduced into the broadcast scheduling procedure. Further, to simplify the encoding/decoding procedure and to minimize the encoding/decoding delay, we apply the XOR-based network coding for its easy implementation, low cost, and short delay, compared with the traditional randomized linear network coding. The pseudocode of the proposed algorithm is given in Algorithm 1. We calculate the scheduling weight of each newly arrived multi-item request and insert it in descending order of the scheduling weight into a service queue maintained at the server. We then select the data item with the most absolute profit in the first pending request Q^* as the encoding seed d_j . After that, a maximal clique which contains seed d_j is searched on the instantly decodable network coding graph G , which is constructed under the conditions given in Section 4.2.2. Next, the server encodes all data items that are included in the founded maximal clique and broadcasts the encoded packet to all clients. Finally, the clients update their caches by decoding their requested data items from the broadcast packet, and the server refreshes the service queue by removing all satisfied requests and expired requests which missed their deadlines.

Given the most profitable request Q^* which has the maximum scheduling weight $\text{PVC}(Q^*)$ and the constructed coding graph G , the proposed NC-based scheduling algorithm searches the maximal cliques that contain the data item with the maximal absolute profit in Q^* over G . Once the maximal clique is found (if multiple maximal cliques are found, chose one randomly), the data items which are recognized by the vertices in the returned clique are encoded into a NC packet, and this NC packet will be broadcasted to all clients. In this way, not only the advantage of PVC algorithm by broadcasting the most profitable data item is kept, but also the opportunity cost is minimized, since

the proposed algorithm makes as much sufficient use of the channel bandwidth as possible. The network coding gain is achieved by using the most weighted data item which has the maximal absolute profit in Q^* as the seed and encoding as much as possible data items requested by multiple clients without interrupting the instantly decodable property, and it in turn compensates the performance loss caused by PVC's scheduling decision to broadcast the most profitable item instead of other items. In particular, if the maximal clique returned by the proposed algorithm contains the item which has the maximal item expected revenue and does not belong to Q^* , then $C(Q) = 0$, which denotes the case where no opportunity cost shall be paid for scheduling request Q^* .

5. Performance Evaluation

In this section, we set up a simulation environment as described in Section 3 and conduct comprehensive experiments to evaluate the performance of the proposed NC-based PVC scheduling algorithm (NPVC (for simplicity, we use NPVC to denote the proposed algorithm in the following)). To better understand the characteristics of NPVC, we implement the well-known algorithms, EDF and SIN- α , both in coding and in noncoding versions, along with the original PVC algorithm, to compare the deadline miss ratio, as well as the ratio of ineffective service over effective service.

5.1. Performance Metrics. The performance metrics used in the simulations are defined as follows:

$$\begin{aligned} \text{deadline miss ratio} &= \frac{\text{total missed requests}}{\text{total requests}}, \\ \text{ratio of ineffective service over effective service} &= \frac{\text{total ineffective num}}{\text{total satisfied num}}. \end{aligned} \quad (8)$$

The deadline miss ratio is the most important performance metric for real-time on-demand data scheduling that it is a faithful metric to describe the system performance of serving requests timely. The ratio of ineffective service over effective service is designed for further examination of performance improvement and interpretation of variations of request deadline miss ratios of different algorithms in simulations. The ineffective service means the service for the requests that are served partially and whose deadlines are missed finally, and the effective service means the service for the requests satisfied at last. Thus, the ratio of ineffective service over effective service is more of a relevant performance metric related to the deadline miss ratio. Given a small deadline miss ratio, which equals the fact that the number of requests that missed their deadlines is small, a lower ratio of ineffective service over effective service indicates that fewer instances of ineffective service were provided for satisfying requests with time constraints and therefore defines a good scheduling algorithm. However, no conclusion can be made with ratio of ineffective service over effective service alone.

Since EDF and SIN- α are designed for single-item requests originally, for the multi-item requests environment,

TABLE 1: System parameter settings.

Parameter	Default	Range	Description
f	8	1-10	Request arrival rate scaling factor
θ	0.5	0-1	Zipf distribution parameter
N	10	10-100	Number of clients
M	40	40-120	Number of data items
\bar{r}	2	1-8	Mean of items in a request
c	10	5-25	Cache size
LAX_MIN	10	1-20	Request deadline scaling factor
LAX_MAX	15	5-25	Request deadline scaling factor

EDF is modified to serve the most urgent request first (i.e., the request with the earliest deadline), and a data item is chosen randomly from the scheduled request to be the encoding seed for network coding-based EDF (NEDF). After that, NEDF searches the optimal coding combination over the instantly decodable coding graph G and broadcasts the encoded items set. Similarly, SIN- α is altered to schedule the data item with the smallest SIN-value first; SIN-value of an item is defined as the ratio of slack time of its parent request to its item productivity. For the network coding-based SIN (NSIN), the data item with the smallest SIN-value is used as the encoding seed and the optimal coding set is searched over the coding graph G and broadcasted to all clients.

5.2. Experimental Setup. The simulation model is based on the system architecture described in Section 3. The main parameters and their settings for the experiments are shown in Table 1 and these default values are used in various simulations, unless stated otherwise. Besides, we use SIN to denote SIN- α algorithm with $\alpha = 1$ in simulations.

In the simulation system, each client generates a stream of requests one by one with exponentially distributed interarrival time. The mean request arrival rate is set to be a constant divided by f , which is actually a scaling factor of the request arrival rate. In the following simulations, the mean request interarrival time for each client is set to be $10/f$ time slots, and the overall mean request interarrival time of the system is $10/(N * f)$, where N is the number of clients.

The number of items in a request follows the uniform distribution in $[\bar{r}-1, \bar{r}+1]$ with expectation \bar{r} . When $\bar{r} = 1$, all requests become single-item requests. Deadline Dl_i of request Q_i is set according to formula (9), where r_i is the original number of items in the request and AT_i is the arrival time of request Q_i . uniform produces a random number selected uniformly from the range of request deadline scaling factor LAX_MIN~LAX_MAX. Consider

$$Dl_i = AT_i + (1 + \text{uniform}(LAX_MIN, LAX_MAX)) \cdot r_i. \quad (9)$$

The data access pattern is shaped with the Zipf distribution with skewness parameter θ , $0 \leq \theta \leq 1$. The smaller the value of θ is, the less different items are accessing frequency among all data items in the database. In particular, when $\theta = 0$, the distribution becomes the uniform distribution,

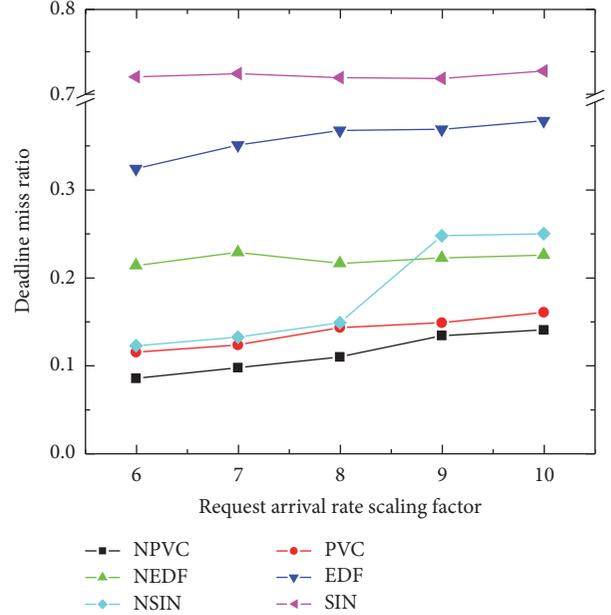


FIGURE 4: Deadline miss ratio with different request arrival rates.

while the Zipf distribution becomes extremely skewed as θ increases to 1. Upon receiving a requested data item, a client stores the item in its local cache, and the LRU (Least Recently Used) policy is used for cache replacement.

5.3. Results Analysis. In this section, we present the results of performance evaluation of different scheduling algorithms for on-demand multi-item requests broadcast system. The results are obtained when the system was in a steady state and all data points are based on the average of over 5000 simulation runs.

5.3.1. Request Arrival Rate Scaling Factor. Figure 4 depicts the deadline miss ratio under different scaling factor f . f is actually the control factor of system load; larger f indicates heavier load and may cause large deadline miss ratio. It can be concluded that the deadline miss ratio increases with f for all compared algorithms. Besides, the adoption of network coding technique reinforces the capabilities of original scheduling algorithms to deliver data items on time, as the deadline miss ratios of NPVC, NEDF, and NSIN are smaller than those of PVC, EDF, and SIN, respectively. Moreover, the proposed NPVC algorithm achieves the smallest deadline miss ratio among all compared algorithms.

Figure 5 depicts the ratio of ineffective service over effective service with different request arrival rates. It can be seen clearly that the ratios of ineffective service over effective service of all compared algorithms increase with the request arrival scaling factor f , and the proposed NPVC has the lowest ratio of ineffective service over effective service among all algorithms.

5.3.2. Skewness Parameter. The skewness parameter θ is a metric to measure the popularity of data items. With θ getting

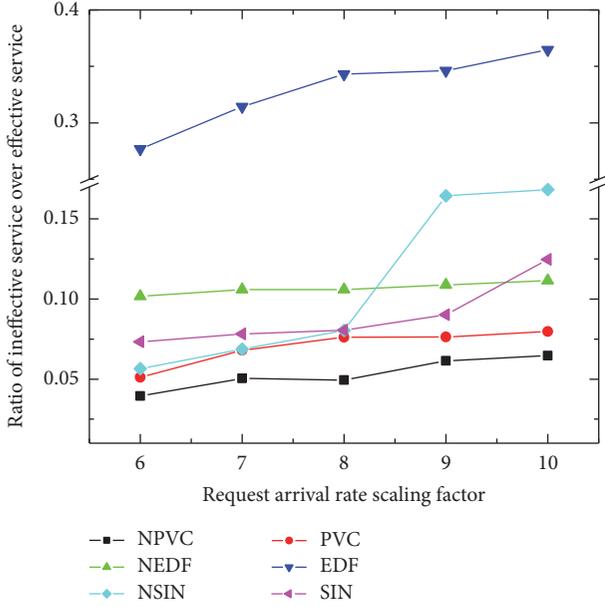


FIGURE 5: Ratio of ineffective service over effective service with different request arrival rates.

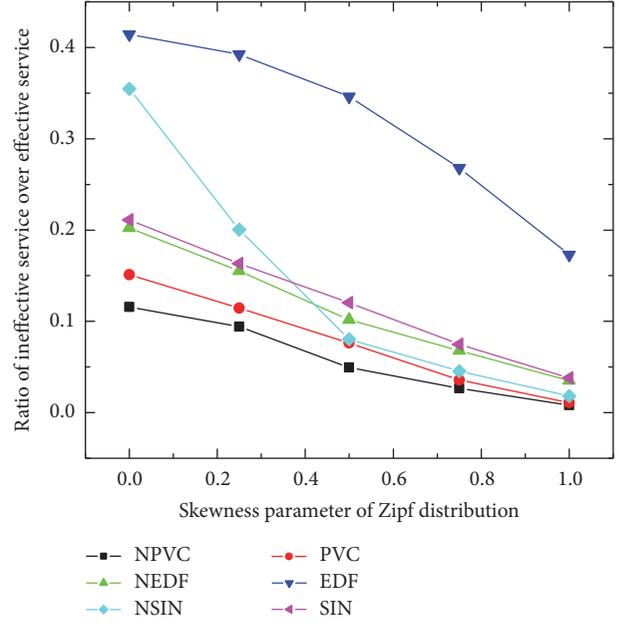


FIGURE 7: Ratio of ineffective service over effective service with different skewness parameters of Zipf distribution.

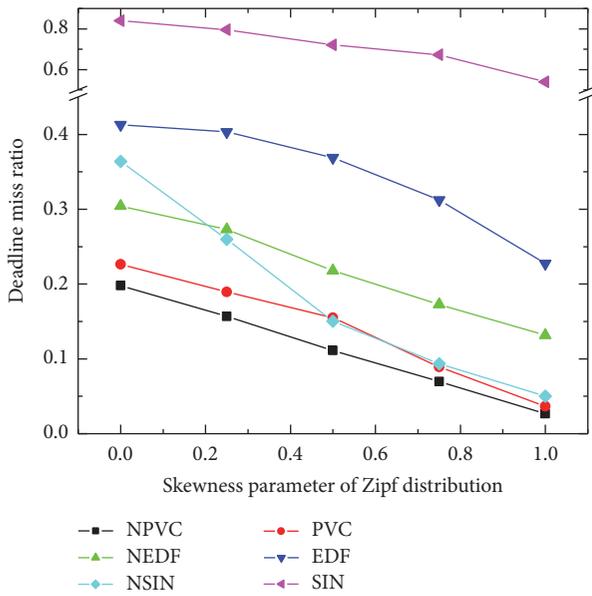


FIGURE 6: Deadline miss ratio with different skewness parameters of Zipf distribution.

closer to 1, the probability of clients requesting “hot” items rises quickly and vice versa. In Figure 6, it can be seen that all algorithms’ deadline miss ratios decrease with the increment of θ . When the data access pattern follows the uniform distribution, namely, $\theta = 0$, the deadline miss ratios are relatively high for all compared algorithms. When $\theta = 1$, NPVC achieves very good performance that its deadline miss ratio is smaller than 3%. Meanwhile, NPVC always has the smallest deadline miss ratio upon different θ .

Figure 7 shows the ratio of ineffective service over effective service under different skewness parameter of Zipf distribution. In general, the ratios of ineffective service over effective service of all algorithms decrease linearly with the increase of skewness parameter. Similarly, the ratio of NPVC is the smallest among all algorithms, indicating its efficiency in fulfilling clients’ requests completely and utilizing wireless bandwidth efficiently.

5.3.3. Deadline Range. Figure 8 shows the simulation results for different range of request deadline scaling factor (LAX_MIN~LAX_MAX). When the mean value of the range of request deadline scaling factor increases, the request deadline miss ratio decreases. The proposed NPVC achieves the best performance under different range of deadline scaling factor. In addition, when the deadline range increases to larger than 15–20, the deadline miss ratios of both PVC and NPVC are less than 1%, which means that the number of requests that missed their deadlines becomes very seldom.

Figure 9 shows the ratio of ineffective service over effective service under different deadline range. The ratios of ineffective service over effective service of all algorithms decrease with deadline range increases, which indicates that the compared algorithms achieve better performance and serve request more efficiently with larger deadline range. When the requests’ deadlines are smaller than 10, the ratio of ineffective service over effective service of algorithm SIN seems to be lower than that of NPVC. However, considering the fact in Figure 8 that SIN has the highest deadline miss ratio and that nearly 77% requests missed their deadlines, the reason that SIN gets the lower ratio is that most of the requests missed their deadlines completely, without any item being transmitted, and thus the requests which were partly

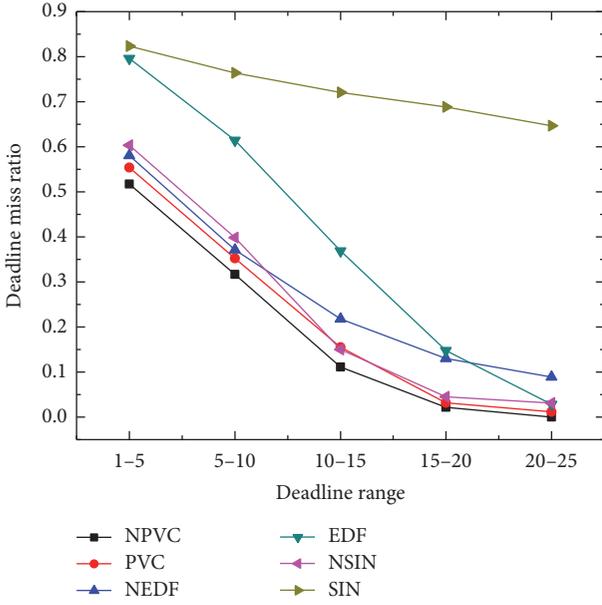


FIGURE 8: Deadline miss ratio with different deadline range.

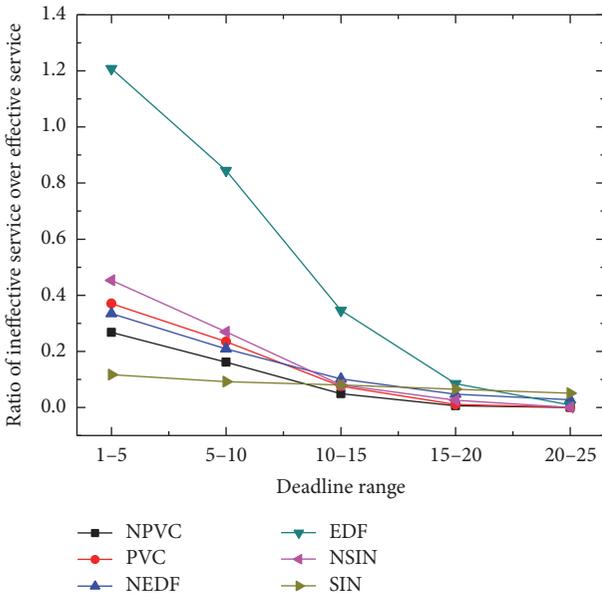


FIGURE 9: Ratio of ineffective service over effective service with different deadline range.

transmitted are rare and cause the low ratio of ineffective service over effective service. In fact, the high deadline miss ratio and low ratio of ineffective service over effective service of SIN confirm its inefficiency in scheduling multi-item requests timely. Contrarily, the proposed NPVC algorithm achieves outstanding performance in both aspects of deadline miss ratio and ratio of ineffective service over effective service, proving its efficiency improvement.

5.3.4. Cache Size. Figure 10 presents the simulation results of deadline miss ratio under different cache size. While the

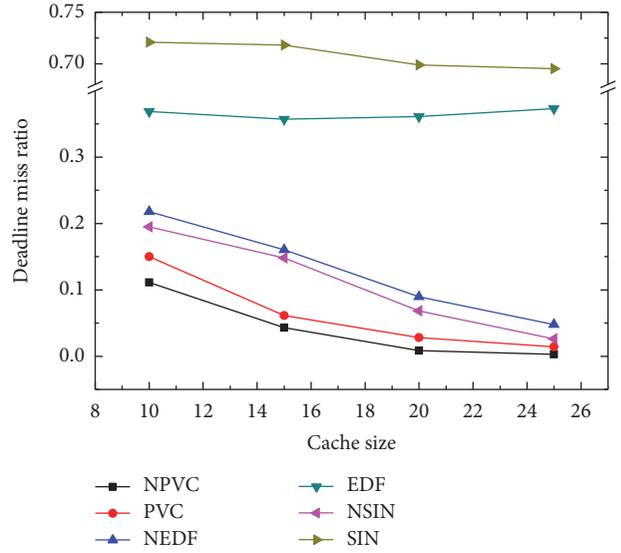


FIGURE 10: Deadline miss ratio with different cache size.

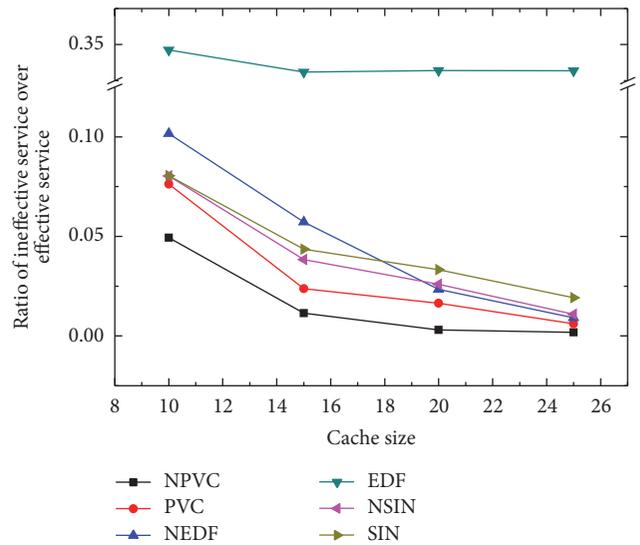


FIGURE 11: Ratio of ineffective service over effective service with different cache size.

cache size increases, more coding opportunities among the requested data items emerge and more requests can meet their deadline requirements. Moreover, when the cache size increases to 20, under the given parameter settings, the system with NPVC achieves impressive performance that the deadline miss ratio is reduced below 1%.

Similarly, NPVC gets the lowest ratio of ineffective service over effective service in Figure 11, and when the cache size becomes larger than 20, the ratio of NPVC gets very close to 0, which implies that little request is partly served and nearly all requests are satisfied successfully (considering the deadline miss ratios together).

6. Conclusions

We have proposed an efficient network coding-based data scheduling algorithm for multi-item requests on-demand data dissemination with time constraints. The primary goal is to minimize the deadline miss ratio and to increase the number of effective services, and this is achieved by integrating network coding with the data scheduling method that broadcasts the most valuable item belonging to the request with the largest “profit.” The concept “profit” describes the performance gain which can be obtained by scheduling the corresponding request first, and the “opportunity cost” represents the gain lost by making the scheduling decision. By introducing network coding into the data scheduling process, the scheduling decision can be made by searching a maximal clique which meets certain rules over a specially constructed graph. In this way, the “opportunity cost” can be alleviated (or removed) and the deadline miss ratio is minimized.

Competing Interests

The authors declare that they have no competing interests.

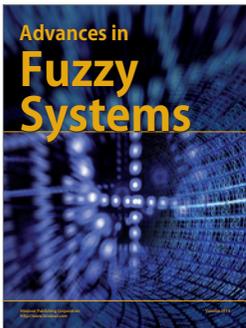
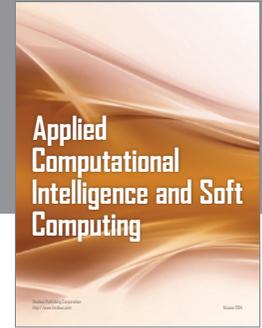
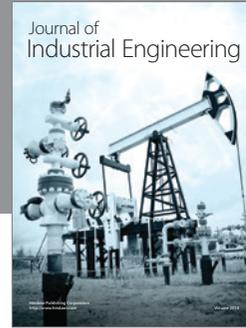
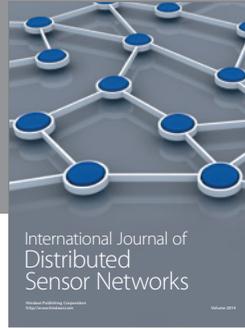
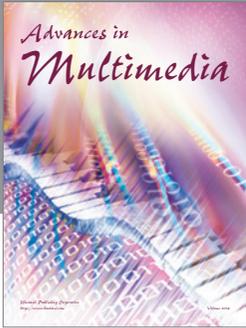
Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant no. 61561029) and funded by the Major Project of Education Department of Yunnan Province (Grant no. 2014Z022) and the Talent Development Project of Kunming University of Science and Technology (Grant no. KKSYS201403024).

References

- [1] Y. Zhu, L. Song, S. Wu, H. Wang, and C. Wang, “Cooperative stepwise relaying and combining for multihop vehicular wireless communication,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2663–2671, 2015.
- [2] F. Xia, Q. Yang, J. Li, J. Cao, L. Liu, and A. M. Ahmed, “Data dissemination using interest-tree in socially aware networking,” *Computer Networks*, vol. 91, pp. 495–507, 2015.
- [3] S. Wang, X. Wang, J. Huang, R. Bie, and X. Cheng, “Analyzing the potential of mobile opportunistic networks for big data applications,” *IEEE Network*, vol. 29, no. 5, pp. 57–63, 2015.
- [4] T. H. Luan, R. Lu, X. Shen, and F. Bai, “Social on the road: enabling secure and efficient social networking on highways,” *IEEE Wireless Communications*, vol. 22, no. 1, pp. 44–51, 2015.
- [5] M. H. Amerimehr and F. Ashtiani, “Performance analysis of network coding-based content distribution in vehicular ad-hoc networks,” *IET Communications*, vol. 8, no. 9, pp. 1447–1458, 2014.
- [6] I.-H. Chuang, T.-A. Lin, C.-C. Lo, and Y.-H. Kuo, “Time-sensitive data dissemination in opportunistic networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '14)*, pp. 2420–2425, Istanbul, Turkey, April 2014.
- [7] M. Dai, K. W. Shum, and C. W. Sung, “Data dissemination with side information and feedback,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 4708–4720, 2014.
- [8] M. H. Firooz and S. Roy, “Data dissemination in wireless networks with network coding,” *IEEE Communications Letters*, vol. 17, no. 5, pp. 944–947, 2013.
- [9] G. Li, Q. Zhou, and J. Li, “A novel scheduling algorithm for supporting periodic queries in broadcast environments,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2419–2432, 2015.
- [10] K. Han, J. Luo, L. Xiang, M. Xiao, and L. Huang, “Achieving energy efficiency and reliability for data dissemination in duty-cycled WSNs,” *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1041–1052, 2015.
- [11] H. Feng, Z. Chen, and H. Liu, “On the push-based converged network with limited storage,” in *Proceedings of the IEEE International Conference on Communications (ICC '15)*, pp. 4474–4479, London, UK, June 2015.
- [12] C. Yang, Z. Chen, Y. Yao, and B. Xia, “Performance analysis of wireless heterogeneous networks with pushing and caching,” in *Proceedings of the IEEE International Conference on Communications (ICC '15)*, pp. 2190–2195, London, UK, June 2015.
- [13] S. Mavridopoulos, P. Nikipolitis, G. Papadimitriou, and P. Sarigiannidis, “Broadcast levels: efficient and lightweight schedule construction for push-based data broadcasting systems,” *IEEE Transactions on Broadcasting*, vol. 61, no. 3, pp. 470–481, 2015.
- [14] W. Sun, Y. Qin, J. Wu et al., “Air indexing for on-demand XML data broadcast,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1371–1381, 2014.
- [15] C.-M. Liu and T.-C. Su, “Broadcasting on-demand data with time constraints using multiple channels in wireless broadcast environments,” *Information Sciences*, vol. 242, pp. 76–91, 2013.
- [16] P. He, H. Shen, and H. Tian, “On-demand data broadcast with deadlines for avoiding conflicts in wireless networks,” *Journal of Systems and Software*, vol. 103, pp. 118–127, 2015.
- [17] W. Hu, C. Fan, J. Luo, C. Peng, and B. Du, “An on-demand data broadcasting scheduling algorithm based on dynamic index strategy,” *Wireless Communications & Mobile Computing*, vol. 15, no. 5, pp. 947–965, 2015.
- [18] H. Dykeman, M. Ammar, and J. Wong, “Scheduling algorithms for videotex systems under broadcast delivery,” in *Proceedings of the IEEE International Conference on Communications*, Toronto, Canada, June 1986.
- [19] D. Aksoy and M. Franklin, “R × W: a scheduling approach for large-scale on-demand data broadcast,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 846–860, 1999.
- [20] W. Hu, C. Xia, B. Du, and M. Wu, “An on-demand data broadcasting scheduling considering the data item size,” *Wireless Networks*, vol. 21, no. 1, pp. 35–56, 2015.
- [21] P. Xuan, S. Sen, O. Gonzalez, J. Fernandez, and K. Ramamritham, “Broadcast on demand: efficient and timely dissemination of data in mobile environments,” in *Proceedings of the 3rd Real-Time Technology and Applications Symposium*, pp. 38–48, IEEE, Montreal, Canada, June 1997.
- [22] J. Xu, X. Tang, and W.-C. Lee, “Time-critical on-demand data broadcast: algorithms, analysis, and performance evaluation,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 1, pp. 3–14, 2006.
- [23] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

- [24] J. Lv, V. C. S. Lee, M. Li, and E. Chen, "Profit-based scheduling and channel allocation for multi-item requests in real-time on-demand data broadcast systems," *Data & Knowledge Engineering*, vol. 73, pp. 23–42, 2012.
- [25] S. Sorour and S. Valaee, "An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 869–878, 2011.
- [26] S. Sorour and S. Valaee, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–5, Cape Town, South Africa, May 2010.
- [27] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1553–1567, 2015.
- [28] S. Acharya and S. Muthukrishnan, "Scheduling on-demand broadcast: new metrics and algorithms," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, pp. 43–54, Dallas, Tex, USA, 1998.
- [29] M. A. Sharaf and P. K. Chrysanthis, "On-demand broadcast: new challenges and scheduling algorithms," in *Proceedings of the 1st Hellenic Data Management Symposium (HDMS '02)*, 2002.
- [30] S.-R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [31] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.
- [32] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Broadcasting with side information: bounding and approximating the broadcast rate," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5811–5823, 2013.
- [33] Y.-C. Kao, C.-N. Lee, P.-J. Wu, and H.-H. Kao, "A network coding equivalent content distribution scheme for efficient peer-to-peer interactive VoD streaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 985–994, 2012.
- [34] Y. Zhou, Y. Xu, and S. Zhang, "Exploring coding benefits in CDN-based VoD systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 11, pp. 1969–1981, 2014.
- [35] I.-H. Hou, "Broadcasting delay-constrained traffic over unreliable wireless links with network coding," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 728–740, 2015.
- [36] J. Chen, V. C. Lee, K. Liu, G. G. Ali, and E. Chan, "Efficient processing of requests with network coding in on-demand data broadcast environments," *Information Sciences*, vol. 232, pp. 27–43, 2013.
- [37] C. Zhan, V. C. S. Lee, J. Wang, and Y. Xu, "Coding-based data broadcast scheduling in on-demand broadcast," *IEEE Transactions on Wireless Communications*, vol. 10, no. 11, pp. 3774–3783, 2011.
- [38] Y. Meng, E. Chan, and V. Lee, "Performance simulation of network coding-based on-demand broadcast models," in *Proceedings of the 22nd Wireless and Optical Communication Conference (WOCC '13)*, Q. Chen, Q. Zhou, and R. Chai, Eds., pp. 109–114, Chongqing University of Posts and Telecommunications, Chongqing, China, May 2013.
- [39] S. Wang and M. Liang, "An efficient multicast strategy for on-demand data dissemination in wireless networks," *Journal of Internet Technology*, vol. 15, no. 4, pp. 557–564, 2014.
- [40] X. Wang, C. Yuen, and Y. Xu, "Coding-based data broadcasting for time-critical applications with rate adaptation," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2429–2442, 2014.
- [41] S. Athanasiadou, M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Xor-based encoding with instantaneous decoding for the broadcast erasure channel with feedback: the three-user case," *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 5274–5287, 2014.
- [42] Z. Zhou and L. Zhou, "Encoded packet-assisted retransmission for multiple unicast flows in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3409–3422, 2013.
- [43] Z. Wang and M. Hassan, "Blind XOR: low-overhead loss recovery for vehicular safety communications," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 35–45, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

