Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Performance analysis and optimization for chunked network coding based wireless cooperative downloading systems*

Xiu-xiu WEN[†‡1], Hui-qiang WANG[1], Jun-yu LIN[2], Guang-sheng FENG[1], Hong-wu LV[1], Ji-zhong HAN[2]

([1]College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

([2]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China)

[†]E-mail: wenxiuxiu@hrbeu.edu.cn

Received June 21, 2016; Revision accepted Nov. 30, 2016; Crosschecked Oct. 31, 2017

Abstract: Dense network coding (NC) is widely used in wireless cooperative downloading systems. Wireless devices have limited computing resources. Researchers have recently found that dense NC is not suitable because of its high coding complexity, and it is necessary to use chunked NC in wireless environments. However, chunked NC can cause more communications, and the amount of communications is affected by the chunk size. Therefore, setting a suitable chunk size to improve the overall performance of chunked NC is a prerequisite for applying it in wireless cooperative downloading systems. Most of the existing studies on chunked NC focus on centralized wireless broadcasting systems, which are different from wireless cooperative downloading systems with distributed features. Accordingly, we study the performance of chunked NC based wireless cooperative downloading systems. First, an analysis model is established using a Markov process taking the distributed features into consideration, and then the block collection completion time of encoded blocks for cooperative downloading is optimized based on the analysis model. Furthermore, queuing theory is used to model the decoding process of the chunked NC. Combining queuing theory with the analysis model, the decoding completion time for cooperative downloading is optimized, and the optimal chunk size is derived. Numerical simulation shows that the block collection completion time and the decode completion time can be largely reduced after optimization.

Key words: Wireless environments; Cooperative downloading; Chunked network coding; Markov process
https://doi.org/10.1631/FITEE.1601361 **CLC number:** TP393.0

1 Introduction

With the development of intelligent transportation systems, the driving experience has been largely improved. However, because of installation costs, wireless access points still cannot be extended to full coverage, and the bandwidth allocated to each vehicle is quite low (Zhou *et al.*, 2014). Cooperative downloading systems can make full use of the limited bandwidth to download files. Vehicles that are in close vicinity can comprise a network as each vehicle downloads parts of the target file, and then shares them with the others through free, energy-efficient, and fast short-range broadcasting. The benefits of cooperation include not only bandwidth savings, but also energy efficiency, throughput enhancement, and cost reductions (Militano *et al.*, 2013).

Random linear network coding (RLNC) is an NC method with low complexity and good expansibility. Applying RLNC in wireless cooperative downloading systems can help the file sharing process combat unreliable wireless environments. Related systems involve CodeTorrent (Lee *et al.*, 2006), VANETCODE (Ahmed and Kanhere, 2006), and CodeOn (Li M *et al.*, 2011). RLNC mixes all packages together to increase the system throughput, and thus it is also called 'dense NC' (Maymounkov *et al.*, 2006). However, by implementing dense NC in real systems, Wang and Li (2006) proved that decoding

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (Nos. 61370212, 61402127, and 61502118)
ORCID: Xiu-xiu WEN, http://orcid.org/0000-0003-4668-4228

rates
decrease dramatically with the increase of the file size. The existing solutions can be classified into two types: decreasing the size of the coding field or decreasing the range for mixing the coding (Magli *et al*., 2013). The first type can decrease the complexity of each coding calculation, e.g., systematic binary RLNC (SB-RLNC) (Heide *et al*., 2009). The second type can decrease the size of the coding matrices, such as chunked NC (also called 'generation-based NC') (Chou *et al*., 2003).

Chunked NC divides the target file into multiple chunks, and codes only within each chunk. There are some variants of chunked NC, such as round-robin chunked NC (Abdelrahman and Gelenbe, 2009; Li Y *et al*., 2012) and overlapped chunked NC (Heidarzadeh and Banihashemi, 2010; Li Y *et al*., 2011). Round-robin chunked NC schedules the chunks in a round-robin way, and overlapped chunked NC allows each chunk to overlap with adjacent chunks. These variants are more suitable for centralized broadcasting systems (Joshi and Soljanin, 2013). Thus, we focus on non-overlapped chunked NC with random scheduling. In this case, dense NC and the baseline, where NC is not applied (Pyattaev *et al*., 2015), are two extremes. The performance of chunked NC falls between dense NC and the baseline. The throughput of chunked NC probably tends to the baseline without a careful setting, resulting in massive extra communications and a long download completion time. Ultimately, setting a suitable chunk size to find the tradeoff between throughput and coding complexity is very important in applying chunked NC in practice.

From the perspective of performance analysis of chunked NC, the earliest research is Maymounkov *et al*. (2006). By modeling block disseminations (from a source without cooperation) as a continuous-time trellis, Maymounkov *et al*. (2006) compared dense NC with chunked NC in terms of coding complexity and communication complexity on wired networks. They showed that chunked NC has extra communication overheads of $O(k/\ln^{1/4}k)$, where k is the number of blocks. Heidarzadeh and Banihashemi (2012) also focused on wired networks, and explored further based on the work of Maymounkov *et al*. (2006). Li Y *et al*. (2011) extended the research in chunked NC analysis to wireless networks, and studied it in centralized broadcasting systems, where the decode delay is derived based on 'coupon collector's brotherhood problem'.

The above theoretical studies provide guidance for applying chunked NC into wireless environments, but there are still some problems to be addressed. On the one hand, the existing studies focus on centralized broadcasting systems, and the research into wireless cooperative downloading systems is quite limited. In a centralized broadcasting system, the center holds the entire file, and broadcasts the file to the nodes around it, but in a cooperative downloading system, each user has only parts of the target file. Users take turns to broadcast the encoded blocks (Zhang *et al*., 2007). Each user can decode the file after collecting enough useful encoded blocks. Therefore, the efficiency of block collection is very different for two systems. On the other hand, the analysis work on (useful encoded) block collection completion time is rough, and analytical work on decode completion time has not been conducted yet. These two metrics can directly affect the service experience in cooperative downloading. The block collection completion time is an important factor determining the availability of the target file, and the decode completion time determines the waiting time for users. If the block collection completion time is too long, users will probably leave the system with an incomplete target file, which impacts the availability of the file. Therefore, optimization of the two metrics is significant for improving the service experience with cooperative downloading systems. Accordingly, we focus on the (encoded) block-sharing process in cooperative downloading systems, and optimize the block collection completion time and decode completion time in this study. Our contributions are as follows:

1. The block sharing process is analyzed, and a system analysis model is proposed based on a Markov process. Considering that users take turns to broadcast, the lower bound for its effect on the efficiency of the collecting process is derived. Furthermore, by applying the lower bound in the model, an approximate formula for the decode delay, i.e., the number of steps required for completing the collection, is obtained.

2. The block collection completion time is determined not only by the decode delay, but also by the length of the steps. When the size of the target file is given, the decode delay decreases with the increase of

the chunk size, but the average length of the steps increases since the encoding complexity increases, so there must be a sweet spot for the chunk size that minimizes the block collection completion time. The optimal chunk size is derived based on this observation. Numerical simulation shows that the block collection completion time can be largely reduced using the optimal chunk size.

3. Queuing theory is used to model the decoding process, and then the queue is combined with the system analysis model to obtain the decode completion time. Furthermore, a calculation method is presented to minimize the decoding completion time. Numerical simulation shows that the decode completion time can be greatly decreased using our calculation method.

2 System model for chunked NC based wireless cooperative downloading

When a node wants to download a file, it broadcasts a request. Any neighbor receiving the request sends a reply, if it is also interested in the file. Therefore, the nodes that are interested in the file and located in close vicinity comprise a network, and cooperative downloading starts.

Cooperative downloading consists of two parts: downloading and sharing. Let $\mathcal{F}_q$ be the coding field, and q is the size of the coding field. The target file is divided into blocks. The blocks are divided into M chunks. Each chunk has r blocks, each of size B. Each block can be seen as a vector consisting of elements of $\mathcal{F}_q$. Let $\mathcal{F}_q^B$ be the set of these vectors. In the downloading process, each node downloads encoded blocks stochastically. We assume that the target file can be decoded using the blocks downloaded by all nodes (Wang and Lin, 2014), and focus on the sharing process. In this process, time is divided into steps (Li Y *et al.*, 2011), and some nodes are selected as broadcast nodes in each step.

Each broadcast node chooses a chunk and produces an encoded block. Let $\boldsymbol{F}=[\boldsymbol{p}_1\ \boldsymbol{p}_2\ \dots\ \boldsymbol{p}_r]'$ be the matrix consisting of the blocks of the chunk. Each encoded block can be seen as the multiplication of $\boldsymbol{F}$ and a coding vector. Let $\boldsymbol{B}_1, \boldsymbol{B}_2, \dots, \boldsymbol{B}_k$ be the encoded blocks (of the chunk) held by the broadcast node, and $\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_k$ are their corresponding coding vectors. The broadcast node produces the encoded block as $\sum_{i=1}^{k}\alpha_i\boldsymbol{B}_i$ (α_i is stochastically chosen from $\mathcal{F}_q$ according to RLNC), and then broadcasts $\sum_{i=1}^{k}\alpha_i\boldsymbol{B}_i$ with its coding vector $\sum_{i=1}^{k}\alpha_i\boldsymbol{b}_i$. Fig. 1 presents a scenario for cooperative downloading, where M=1, r=4, q=4, k=2, and RSU is a road side unit. Node 1 is selected as a broadcast node. The encoded blocks held by node 1 are $\boldsymbol{p}_3+\boldsymbol{p}_4$ and $\boldsymbol{p}_2+\boldsymbol{p}_4$, and their coding vectors are [0 0 1 1] and [0 1 0 1]. Assuming that the coefficients chosen by node 1 are α_1=1 and α_2=1, then the broadcasted, encoded block is $\boldsymbol{p}_2+\boldsymbol{p}_3+2\boldsymbol{p}_4$, and its coding vector is [0 1 1 2].

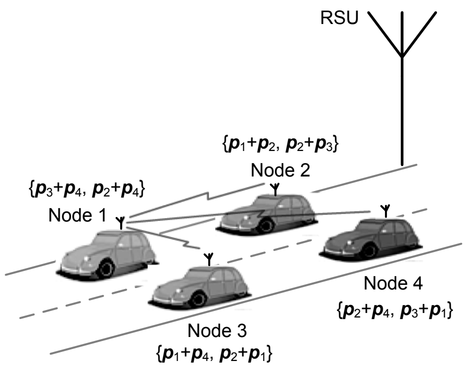

Fig. 1 A cooperative downloading scenario

Some nodes around the broadcast node can receive the encoded block $\sum_{i=1}^{k}\alpha_i\boldsymbol{B}_i$ and its coding vector $\sum_{i=1}^{k}\alpha_i\boldsymbol{b}_i$ successfully. Take one of the nodes for instance. After it receives the encoded block, it checks whether the encoded block is a linear combination of the encoded blocks that it already has. If not, the encoded block is useful; else, the encoded block will be ignored. After the node collects r useful encoded blocks about the chunk, it can decode $\boldsymbol{F}$ from the equation group consisting of the coding vectors and encoded blocks. If all the chunks are decoded, the target file can be obtained. After all nodes decode the target file, cooperative downloading ends.

Dense NC can be seen as a special case of chunked NC with M=1 (Fig. 2). Chunked NC has some advantages compared with dense NC:

1. Decreased coding complexity. Dividing the target file into multiple chunks can decrease the size of the equation group, and then the amount of calcu-

lations for decoding the file. Meanwhile, the complexity of encoding is reduced, and the expected length of the steps is shortened.

2. Reduced decode completion time. Chunks can be decoded individually. When some chunks are decoded, other chunks may still be on their way to collecting encoded blocks. Thus, the decoding process and the collecting process can be carried out simultaneously, and the decode completion time of the target file can be reduced.

However, chunked NC can increase the decode delay, so it is necessary to set a suitable r according to the network situation. The first aim of this work is to establish a model for analyzing the performance of chunked NC in the wireless cooperative downloading system, and the second aim is to optimize the performance based on the model.

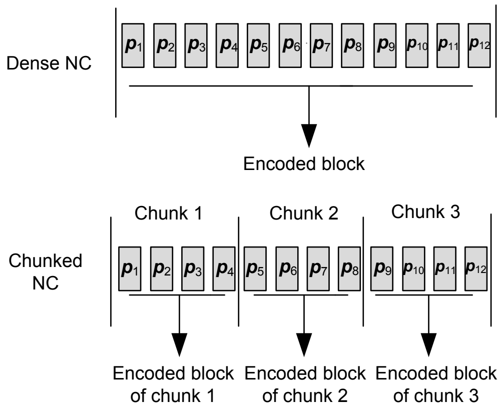

Fig. 2 Dense network coding (NC) versus chunked NC with M=3 and r=4

M and r are the number of chunks and the number of blocks of each chunk, respectively

3 System analysis model

In this section, we propose a system analysis model based on a Markov process, and solve the model using the Monte Carlo method. Additionally, extensive simulations are performed, and an approximate formula for decode delay is obtained by fitting the results of the simulations.

Performance analysis for wireless broadcast networks is very complex. According to the network organization process in Section 2, nodes do not forward requests, so the radius of the network is 2 hops at most. Therefore, the network is similar to the broadcast networks in Lucani *et al*. (2009a; 2009b) and Yu *et al*. (2014). We use their analysis concepts in analyzing a node's performance in the worst case, and then use it to appraise the cooperative downloading system.

Let $(g_1, g_2, \ldots, g_M)$ be the state of a node, where g_i (i=1, 2, …, M) is the number of useful encoded blocks of chunk i collected by the node, including those downloaded by itself and those shared by others. Block collection can be described by a Markov process on the discrete state space of $\{(g_1, g_2, \ldots, g_M)|0\le g_i\le r, g_i\in\mathbb{Z}, 1\le i\le M\}$. In each step, a node can receive an encoded block at most. Let $(g_1, g_2, \ldots, g_M)$ be the current state of the node. If the received encoded block is useful, then the node will go to any state of $\{(g_1, g_2, \ldots, g_i+1, \ldots, g_M)|0\le g_i<r, 1\le i\le M\}$; else, it stays at the current state. For any $0\le g_i<r$, $1\le i\le M$, the probability that the node goes to $(g_1, g_2, \ldots, g_i+1, \ldots, g_M)$ is

$$p_{(g_1,g_2,\cdots,g_M),(g_1,g_2,\cdots,g_i+1,\cdots,g_M)} = p_{\text{rec}} f_{\text{hit}}(i) f_{\text{incr}}(g_i). \quad (1)$$

Here, p_{rec} is the probability that the node is not selected as a broadcast node and it receives an encoded block successfully. $f_{\text{hit}}(i)$ is the probability that the encoded block belongs to chunk i, and $f_{\text{incr}}(g_i)$ is the probability that the encoded block is useful. The probability of staying at the current state is

$$p_{(g_1,g_2,\cdots,g_M),(g_1,g_2,\cdots,g_M)} = 1 - \sum_{g_i<r,1\le i\le M} p_{\text{rec}} f_{\text{hit}}(i) f_{\text{incr}}(g_i). \quad (2)$$

In wireless environments, p_{rec} is affected by many factors, such as the broadcast node selection strategies, block size, hidden terminals, channel conditions, and mobility (Ma and Chen, 2007). There has been considerable work on this topic (Choi *et al*., 2005; Ma *et al*., 2011). Here we assume p_{rec} is a constant, and analyze only $f_{\text{incr}}(g_i)$.

3.1 Probability of receiving useful encoded blocks

Let u and v be two nodes of the network. We assume that u is a broadcast node in a step. u selects chunk i, and broadcasts an encoded block of the chunk. Let $\boldsymbol{S}_u$ and $\boldsymbol{S}_v$ be the vector spaces spanned by the coding vectors (corresponding to encoded blocks of chunk i) of u and v at the beginning of this step, respectively. v receives the encoded block successfully. Let $\dim(\boldsymbol{S}_u)$ be the dimension of $\boldsymbol{S}_u$. According to the description in Section 2, the encoding process is equivalent to the stochastic selection of a vector $\boldsymbol{b}$ from $\boldsymbol{S}_u$. Whether the encoded block is useful for v is equivalent to whether $\boldsymbol{b}$ belongs to $\boldsymbol{S}_v$.

If the current step is the first step, $\boldsymbol{S}_u$ and $\boldsymbol{S}_v$ could be any subspaces of $\mathcal{F}_q^r$ because each node downloads blocks at random before the block sharing process. Therefore, $\boldsymbol{b}$ can be any nonzero vector of $\mathcal{F}_q^r$ with an equal probability for v. Thus, the probability that $\boldsymbol{b}$ is useful for v is

$$P\{\boldsymbol{b}\notin \boldsymbol{S}_u \mid \boldsymbol{b}\in \boldsymbol{S}_v, \boldsymbol{b}\neq \boldsymbol{0}\} = P\{\boldsymbol{b}\notin \boldsymbol{S}_u \mid \boldsymbol{b}\neq \boldsymbol{0}\} = 1-\frac{q^{\dim(\boldsymbol{S}_u)}-1}{q^r-1} = \frac{q^r-q^{\dim(\boldsymbol{S}_u)}}{q^r-1}. \tag{3}$$

In the first step, $\boldsymbol{S}_u$ and $\boldsymbol{S}_v$ could be any subspace of $\mathcal{F}_q^r$ with the dimension of $\dim(\boldsymbol{S}_u)$. However, nodes in close vicinity will receive some common vectors with an increasing number of steps, and $\boldsymbol{S}_u$ is not arbitrary for v in this case.

In the following we analyze the probability of receiving useful blocks when the current step is not the first one. We still assume that u is a broadcast node in the current step, and u selects chunk i, and $\boldsymbol{S}_u$ and $\boldsymbol{S}_v$ are the vector spaces spanned by vectors of chunk i collected by u and v until this step. u and v have some common vectors, including those shared by their common neighbors and those broadcasted by u or v. Denote the vector space spanned by the common vectors as $\boldsymbol{W}$. It is obvious that $\boldsymbol{W}$ is a common subspace of $\boldsymbol{S}_u$ and $\boldsymbol{S}_v$. Let $\boldsymbol{W}_\text{L}$ be the orthogonal complement of $\boldsymbol{W}$ under $\mathcal{F}_q^r$. $\boldsymbol{S}_u'$ ($\boldsymbol{S}_v'$) denotes the intersection between $\boldsymbol{S}_u$ ($\boldsymbol{S}_v$) and $\boldsymbol{W}_\text{L}$. Apparently, $\boldsymbol{S}_u$ is the direct sum of $\boldsymbol{S}_u'$ and $\boldsymbol{W}$. Let $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{\dim(\boldsymbol{S}_u')}$ be a set of bases for $\boldsymbol{S}_u'$, and let $\boldsymbol{u}_{\dim(\boldsymbol{S}_u')+1}, \boldsymbol{u}_{\dim(\boldsymbol{S}_u')+2}, \ldots, \boldsymbol{u}_{\dim(\boldsymbol{S}_u)}$ be a set of bases for $\boldsymbol{W}$. Then, $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{\dim(\boldsymbol{S}_u)}$ is a set of bases for $\boldsymbol{S}_u$. The (coding) vector broadcast by u can be expressed as $\sum_{i=1}^{\dim(\boldsymbol{S}_u)}\alpha_i\boldsymbol{u}_i$, where α_i is selected from $\mathcal{F}_q^r$ stochastically. Let $\boldsymbol{b}$ be the vector. We then separate $\boldsymbol{b}$ into two parts, $\boldsymbol{b}'$ and $\boldsymbol{a}$, where $\boldsymbol{b}'=\sum_{i=1}^{\dim(\boldsymbol{S}_u')}\alpha_i\boldsymbol{u}_i$ and $\boldsymbol{a}=\sum_{i=\dim(\boldsymbol{S}_u')+1}^{\dim(\boldsymbol{S}_u)}\alpha_i\boldsymbol{u}_i$. $\boldsymbol{b}'$ is a vector of $\boldsymbol{S}_u'$, and $\boldsymbol{a}$ is a vector of $\boldsymbol{W}$. $\boldsymbol{b}'$ and $\boldsymbol{a}$ cannot be $\boldsymbol{0}$ at the same time because $\boldsymbol{b}$ is nonzero. The probability that $\boldsymbol{b}$ is useful for v is

$$\begin{aligned}
&P\{\boldsymbol{b}\notin \boldsymbol{S}_v \mid \boldsymbol{b}\in \boldsymbol{S}_u,\ \boldsymbol{b}\neq \boldsymbol{0}\}\\
&=P\{\boldsymbol{b}\notin \boldsymbol{S}_v'\oplus \boldsymbol{W} \mid \boldsymbol{b}\neq \boldsymbol{0},\ \boldsymbol{b}=\boldsymbol{a}+\boldsymbol{b}',\ \boldsymbol{b}'\in \boldsymbol{S}_u',\ \boldsymbol{a}\in \boldsymbol{W}\}\\
&=P\{\boldsymbol{a}+\boldsymbol{b}'\notin \boldsymbol{S}_v'\oplus \boldsymbol{W} \mid \boldsymbol{a}+\boldsymbol{b}'\neq \boldsymbol{0},\ \boldsymbol{b}'\in \boldsymbol{S}_u',\ \boldsymbol{a}\in \boldsymbol{W}\}\\
&=P\{\boldsymbol{b}'\notin \boldsymbol{S}_v' \mid \boldsymbol{a}+\boldsymbol{b}'\neq \boldsymbol{0},\ \boldsymbol{b}'\in \boldsymbol{S}_u',\ \boldsymbol{a}\in \boldsymbol{W}\}\\
&=P\{\boldsymbol{b}'\notin \boldsymbol{S}_v' \mid \boldsymbol{b}'\neq \boldsymbol{0},\ \boldsymbol{a}+\boldsymbol{b}'\neq \boldsymbol{0}\}P\{\boldsymbol{b}'\neq \boldsymbol{0} \mid \boldsymbol{a}+\boldsymbol{b}'\neq \boldsymbol{0}\}\\
&=P\{\boldsymbol{b}'\notin \boldsymbol{S}_v' \mid \boldsymbol{b}'\in \boldsymbol{S}_u',\ \boldsymbol{b}'\neq \boldsymbol{0}\}P\{\boldsymbol{b}'\neq \boldsymbol{0} \mid \boldsymbol{a}+\boldsymbol{b}'\neq \boldsymbol{0}\}.
\end{aligned} \tag{4}$$

A vector is nonzero if and only if not all coordinates of the vector are zero:

$$\begin{aligned}
&P\{\boldsymbol{b}'\neq \boldsymbol{0} \mid \boldsymbol{a}+\boldsymbol{b}'\neq \boldsymbol{0},\ \boldsymbol{b}'\in \boldsymbol{S}_u',\ \boldsymbol{a}\in \boldsymbol{W}\}\\
&=P\left\{\sum_{i=1}^{\dim(\boldsymbol{S}_u')}\alpha_i\boldsymbol{u}_i\neq \boldsymbol{0} \,\middle|\, \sum_{i=1}^{\dim(\boldsymbol{S}_u)}\alpha_i\boldsymbol{u}_i\neq \boldsymbol{0}\right\}\\
&=\frac{P\{\text{At least one of } \boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_{\dim(\boldsymbol{S}_u')} \text{ is nonzero}\}}{P\{\text{At least one of } \boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_{\dim(\boldsymbol{S}_u)} \text{ is nonzero}\}}\\
&=\frac{1-q^{\dim(\boldsymbol{W})-\dim(\boldsymbol{S}_u)}}{1-q^{-\dim(\boldsymbol{S}_u)}}.
\end{aligned} \tag{5}$$

Both $\boldsymbol{S}_v'$ and $\boldsymbol{S}_u'$ are subspaces of $\boldsymbol{W}_\text{L}$. By constructing $\boldsymbol{W}_\text{L}$, the effect of $\boldsymbol{W}$ is removed. Given $\boldsymbol{S}_v'$, $\boldsymbol{S}_u'$ can be any subspace of $\boldsymbol{W}_\text{L}$ in this case, and Eq. (3) can be applied to calculate $P\{\boldsymbol{b}'\notin\boldsymbol{S}_v' | \boldsymbol{b}'\in\boldsymbol{S}_u', \boldsymbol{b}'\neq\boldsymbol{0}\}$:

$$P\{\boldsymbol{b}'\notin \boldsymbol{S}_v' \mid \boldsymbol{b}'\in \boldsymbol{S}_u',\ \boldsymbol{b}'\neq \boldsymbol{0}\}=\frac{q^r-q^{\dim(\boldsymbol{S}_v)}}{q^r-q^{\dim(\boldsymbol{W})}}. \tag{6}$$

Submitting Eqs. (5) and (6) into Eq. (4), we obtain

$$\begin{aligned}
&P\{\boldsymbol{b}\notin \boldsymbol{S}_v \mid \boldsymbol{b}\in \boldsymbol{S}_u,\ \boldsymbol{b}\neq \boldsymbol{0}\}\\
&=\frac{(q^r-q^{\dim(\boldsymbol{S}_v)})(1-q^{\dim(\boldsymbol{W})-\dim(\boldsymbol{S}_u)})}{(q^r-q^{\dim(\boldsymbol{W})})(1-q^{-\dim(\boldsymbol{S}_u)})}\\
&=\frac{q^r-q^{\dim(\boldsymbol{S}_v)}}{q^r}\cdot\frac{q^r(q^{\dim(\boldsymbol{S}_u)}-q^{\dim(\boldsymbol{W})})}{(q^r-q^{\dim(\boldsymbol{W})})(q^{\dim(\boldsymbol{S}_u)}-1)}\\
&>\frac{q^r-q^{\dim(\boldsymbol{S}_v)}}{q^r}\cdot\frac{q^{\dim(\boldsymbol{S}_u)}-q^{\dim(\boldsymbol{W})}}{q^{\dim(\boldsymbol{S}_u)}},
\end{aligned} \tag{7}$$

where $\boldsymbol{W}$ is a subspace of $\boldsymbol{S}_u$, so $\dim(\boldsymbol{S}_u)\geq\dim(\boldsymbol{W})$.

$(q^{\dim(\boldsymbol{S}_u)} - q^{\dim(\boldsymbol{W})}) / q^{\dim(\boldsymbol{S}_u)} > 1 - q^{-1}$ when $\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{W})$, and $(q^{\dim(\boldsymbol{S}_u)} - q^{\dim(\boldsymbol{W})}) / q^{\dim(\boldsymbol{S}_u)} = 0$ if $\dim(\boldsymbol{S}_u) = \dim(\boldsymbol{W})$. Thus, Eq. (7) can be written as

$$P\{\boldsymbol{b} \notin \boldsymbol{S}_v \mid \boldsymbol{b} \in \boldsymbol{S}_u,\ \boldsymbol{b} \neq \boldsymbol{0}\} > (1 - q^{\dim(\boldsymbol{S}_v)-r})(1 - q^{-1}) P\{\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{W})\}. \tag{8}$$

Proposition 1

$$P\{\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{W})\} > [2-(1-q^{-1})^{-2}](1-q^{-1})^{-2}.$$

Proof Let n_u be the number of vectors received by u until this step. According to the description above, some of them are also received by v. Denote the others as $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n_u'}$. Because the receiving sequence does not affect the result of $P\{\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{W})\}$, we assume that u first receives the vectors that are also received by v. Let $\boldsymbol{S}_u''$ be the vector space of u at this time. Then we have $\boldsymbol{S}_u'' = \boldsymbol{W}$. When u receives $\boldsymbol{b}_i$, $1 \le i \le n_u'$, it is a non-broadcast node. Let x_i be the node broadcasting $\boldsymbol{b}_i$, where x_i is a neighbor of u. When x_i broadcasts, let $\boldsymbol{W}_i$ be the common subspace of $\boldsymbol{S}_u''$ and $\boldsymbol{S}_{x_i}$, caused by the vectors received in common. To simplify the description, we create the following definitions: $p_i' = P\{\dim(\boldsymbol{W}_i) < \dim(\boldsymbol{S}_{x_i})\}$, and $p_i'' = P\{\boldsymbol{b} \notin \boldsymbol{S}_u'' | \boldsymbol{b}_i \in \boldsymbol{S}_{x_i}, \boldsymbol{b} \neq \boldsymbol{0}\}$. Substituting Eq. (8) into p_i'', we obtain

$$p_i'' \ge (1-q^{-1})^2 p_i'. \tag{9}$$

Clearly, $\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{S}_u'')$ as long as at least one of $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n_u'}$ satisfies $\boldsymbol{b}_i \notin \boldsymbol{S}_u''$.

$$P\{\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{S}_u'')\} = 1 - \prod_{i=1}^{n_u'} (1 - p_i''). \tag{10}$$

Let p'' and p' be $\min_{1 \le i \le n_u'}\{p_i''\}$ and $\min_{0 \le i \le n_u'}\{p_i'\}$, respectively, where p_0' denotes $P\{\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{W})\}$. Based on Eqs. (9) and (10), we obtain

$$p'' \ge (1-q^{-1})^2 p', \tag{11}$$

$$p' > 1-(1-p'')^{n_u'}. \tag{12}$$

Based on Eq. (11), $p' \le p''(1-q^{-1})^{-2}$ can be obtained. Substituting $p' \le p''(1-q^{-1})^{-2}$ into Eq. (12), it can be derived that $1-(1-p'')^{n_u'} < p''(1-q^{-1})^{-2}$. Assuming $n_u' \ge 2$, we obtain

$$1-(1-p'')^2 < p''(1-q^{-1})^{-2}. \tag{13}$$

Solving Eq. (13), $p'' > 2-(1-q^{-1})^{-2}$ can be obtained. Therefore, we obtain the lower bound for p' based on Eq. (12) as $p' > 1-[(1-q^{-1})^{-2}-1]^{n_u'} > [2-(1-q^{-1})^{-2}] \cdot (1-q^{-1})^{-2}$.

Furthermore, according to $p' = \min_{0 \le i \le n_u'}\{p_i'\}$ and $P\{\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{W})\} > p_0' > p'$, the lower bound of $P\{\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{W})\}$ mentioned in Proposition 1 can be obtained as $P\{\dim(\boldsymbol{S}_u) > \dim(\boldsymbol{W})\} > p' > [2-(1-q^{-1})^{-2}](1-q^{-1})^{-2}$.

Based on Proposition 1 and Eq. (8), we obtain

$$P\{\boldsymbol{b} \notin \boldsymbol{S}_v | \boldsymbol{b} \in \boldsymbol{S}_u, \boldsymbol{b} \neq \boldsymbol{0}\} > (1-q^{\dim(\boldsymbol{S}_v)-r})c. \tag{14}$$

Here $c = [2-(1-q^{-1})^{-2}](1-q^{-1})^{-1}$. Proposition 1 implies that the common vectors held by broadcast nodes and non-broadcast nodes can decrease the efficiency of the block collection process. Note that the above analysis is not for a specific generation. Therefore, the lower bound of $f_{\text{incr}}(g_i)$ can be obtained:

$$f_{\text{incr}}(g_i) \ge (1-q^{g_i-r})c. \tag{15}$$

The transition probability of the model can be calculated as

$$p_{(g_1,g_2,\cdots,g_M),(g_1,g_2,\cdots,g_i+1,\cdots,g_M)} \ge p_{\text{rec}} f_{\text{hit}}(i)(1-q^{g_i-r})c. \tag{16}$$

The lower bound is a decreasing function of q. We can reduce the complexity of coding by decreasing q, but it may increase the communication overheads by a factor. Therefore, we set q as 256 in simulation. Besides, since we consider the worst case, the lower bound of the transition probability is used to solve the model.

3.2 Solution to the model

There are r^M states in the model. The complexity is too high if we solve the model directly. Therefore, we obtain the expected number of steps from state to state by simulating the transition of the model. Denote $t(g_1, g_2, \ldots, g_M)$ as the expected number of steps where the node stays at $(g_1, g_2, \ldots, g_M)$. Because $t(g_1, g_2, \ldots, g_M)$ follows a geometric distribution, we obtain

$$E\left[t(g_1,g_2,\cdots,g_M)\right]=1/(1-p_{(g_1,g_2,\cdots,g_M),(g_1,g_2,\cdots,g_M)})$$
$$=\frac{1}{\sum\limits_{1\le i\le M} p_{\text{rec}} f_{\text{hit}}(i) f_{\text{incr}}(g_i)}. \tag{17}$$

After the node leaves (g_1, g_2, …, g_M), it may go to (g_1, g_2, …, g_i+1, …, g_M) with the probability of θ_i:

$$\theta_i=\frac{p_{(g_1,g_2,\cdots,g_M),(g_1,g_2,\cdots,g_i+1,\cdots,g_M)}}{1-p_{(g_1,g_2,\cdots,g_M),(g_1,g_2,\cdots,g_M)}}$$
$$=\frac{p_{\text{rec}} f_{\text{hit}}(i) f_{\text{incr}}(g_i)}{\sum\limits_{1\le i\le M} p_{\text{rec}} f_{\text{hit}}(i) f_{\text{incr}}(g_i)}=\frac{f_{\text{hit}}(i) f_{\text{incr}}(g_i)}{\sum\limits_{1\le i\le M} f_{\text{hit}}(i) f_{\text{incr}}(g_i)}. \tag{18}$$

We use logic expressions to describe the target states. $b_0: \bigwedge_{1\le i\le M}(g_i=r)$ means that all chunks have finished the collection, and $b_1: \bigvee_{1\le i\le M}(g_i=r)$ means that at least one chunk has finished its collection. Let s=(g_1, g_2, …, g_M) be the start state. Then the number of steps to reach any state satisfying b can be calculated using Algorithm 1.

Algorithm 1 Model solution

Input: start state, s=(g_1, g_2, …, g_M)
target states, b
Output: number of steps from the start state to the target states, D
1 D←0
2 Obtain distribution θ based on s
3 **while** ¬b is true
4 **if** b_0 is true
5 D←∞
6 **break**
7 **end if**
8 D←D+E[t(g_1, g_2, …, g_M)]
9 i←rand(θ)
10 $s.g_i$=$s.g_i$+1
11 Uptate θ
12 **end while**
13 **return** D

In each loop, line 8 first accumulates the number of steps based on Eq. (17), and then lines 9 and 10 stochastically select a state in {(g_1, g_2, …, g_i+1, …, g_M)|0≤g_i<r, 1≤i≤M } according to θ, and the node goes to that state in the next loop. The run ends when b turns true, and the number of steps to obtain the target states can be obtained. There are some states that may not be transited in a run, which makes Algorithm 1 infinite loops. The aim of lines 4–7 is to prevent the infinite loops. The number of loops is $\sum_{i=1}^{M}(r-g_1)$ at most, so the complexity of the algorithm is $O(rM)$. The result of a run is not convincing enough, so a lot of runs are performed for each case, and we use the averaged result to approximate the expected number of steps from the start state to the target states.

The decode delay is the number of steps where all chunks have collected enough useful encoded blocks (Eryilmaz *et al*., 2008). Any chunk can be decoded after it collects enough blocks, so the step when the decoding process starts is where one of the chunks collects enough blocks. We call it the decode start time, which is also very important because it is the basis of decode completion time analysis.

3.2.1 Decode delay

Denote decode delay as f_{total}. The corresponding target states should satisfy $b_0: \bigwedge_{1\le i\le M}(g_i=r)$. Under the random strategy, $f_{\text{hit}}(i)=1/M$. The worst situation is considered here, so the start state is set as $g_1=g_2=\ldots=g_M=0$. According to Eq. (17), E[t(g_1, g_2, …, g_M)] is inversely proportional to p_{rec}, so f_{total} is inversely proportional to p_{rec}, and we set p_{rec} as 1 for simplification. q is set as 256. The ranges of r and M are [20, 100] and [200, 1000], respectively. If B is 1024 bytes, then the corresponding size of the target file is in the range of [4, 100] MB, which can satisfy the regular download needs of the nodes. We run the algorithm 100 times for each case with fixed r and M, and use the averaged result to approximate f_{total}.

Figs. 3 and 4 show that f_{total} is proportional to M and r. In the simplest case where M=1, it is clear that $f_{\text{total}}=\sum_{i=1}^{r}1/(1-q^{i-r})c$, and f_{total} increases linearly with r if r is higher than a threshold, such as 1−log(1−ε) (ε is a very small number approximating 0). When M>1, the number of steps to collect enough blocks is $\sum_{i=1}^{r}M/(1-q^{i-r})c$ if we observe from the perspective of a single chunk. Therefore, f_{total} is proportional to M and r, and we can use $\hat{f}_{\text{total}}=M(c_{11}r+c_{12})+c_2$ to approximate f_{total}. Accordingly, a number of simulations are performed, and c_{11}=1.201, c_{12}=17.59, c_2=−1632 are obtained by fitting.

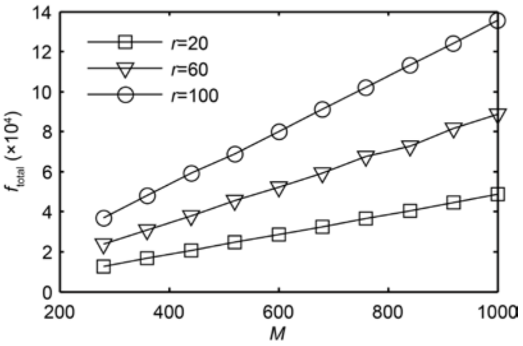

Fig. 3 Effect of M on f_{total} with three different values of r

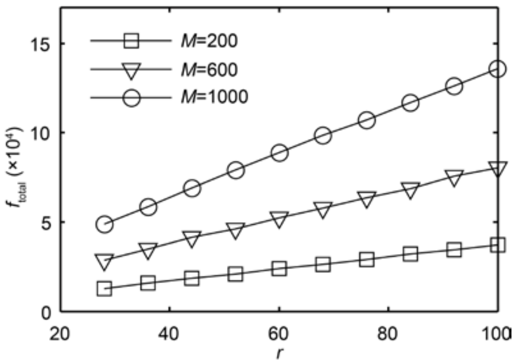

Fig. 4 Effect of r on f_{total} with three different values of M

Fig. 5 presents the estimation errors of $\hat{f}_{total}$ as $\left|\hat{f}_{total} - f_{total}\right| / f_{total}$. It is shown that the estimation errors are less than 3%, so $\hat{f}_{total}$ can approximate f_{total} well. The estimation errors decrease with the increase of r. If $M >> r$, f_{total} is approximately proportional to $M \log M$ (Newman, 1960). Therefore, f_{total} can be approximated by $\hat{f}_{total}$ only when M is not much larger than r. When r=75 and 100, the difference between M and r is larger than that in the cases of r=20 and 25, so $\hat{f}_{total}$ can approximate f_{total} better with the increase of r.

Above all, $\hat{f}_{total}$ can be written as follows when considering p_{rec}:

$$\hat{f}_{total} = [M(c_{11}r + c_{12}) + c_2] / p_{rec}. \tag{19}$$

3.2.2 Decode start time

Denote the decode start time as f_{first}. The corresponding target states should satisfy $b_1 : \underset{1 \le i \le M}{\vee} (g_i = r)$. Other settings are the same as in the previous section. Figs. 6 and 7 present the effect of M and r on f_{first}. f_{first} increases linearly with M and r. Similar to the previous section, we obtain an approximate formula of f_{first} by fitting:

$$\hat{f}_{first} = [M(c'_{11}r + c'_{12}) + c'_2] / p_{rec}, \tag{20}$$

where c_{11}'=0.7991, c_{12}'=−10.3, c_2'=984.4.

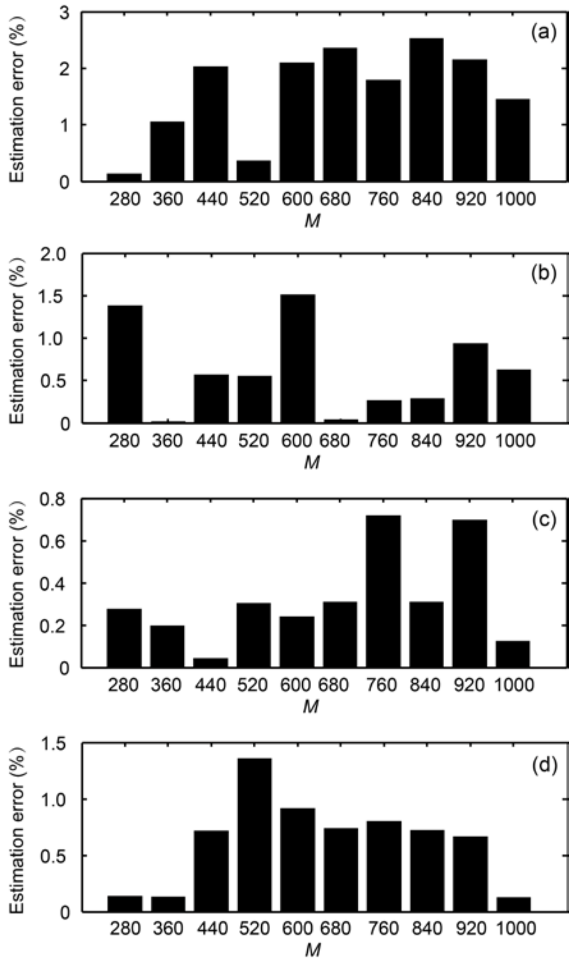

Fig. 5 Estimation errors of $\hat{f}_{total}$: (a) r=20; (b) r=25; (c) r=75; (d) r=100

The estimation errors of $\hat{f}_{first}$ are presented in Fig. 8, which are less than 4%, and decrease with the increase of r. We found that f_{first} is a constant when r is fixed and M>>r. Therefore, f_{total} increases only linearly with r and M when M is not much larger than r. Thus, the accuracy of $\hat{f}_{first}$ decreases with the increase of M and increases with the increase of r.

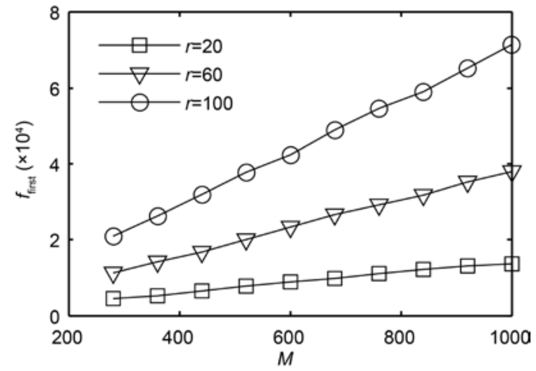

Fig. 6 Effect of M on f_{first} with three different values of r

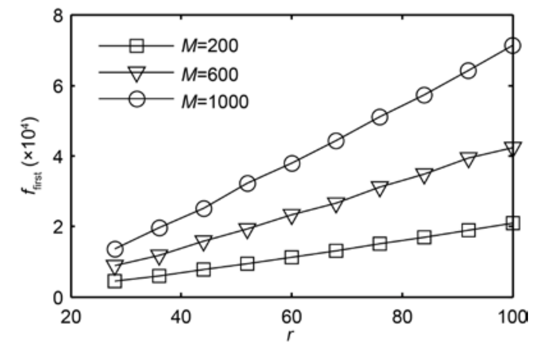

Fig. 7 Effect of r on f_{first} with three different values of M

Let S=rM be the size of the target file. If S is given, f_{first} and the complexity of the coding (encoding and decoding) increase with r, but f_{total} decreases. f_{first} and f_{total} can then affect the block collection completion time and the decode completion time, which will be analyzed in the next section.

4 Performance optimization

In this section, we focus on performance optimization using the approximate formula derived in the previous section. We first try to find the optimal r for the block collection completion time taking into consideration the encoding complexity and wireless broadcasting features, and then we obtain the calculation method to optimize the decoding completion time by modeling the decoding process using queuing theory. We also conduct a numerical simulation, and the simulation results show that the block collection completion time and decode completion time can be largely reduced using our optimization methods.

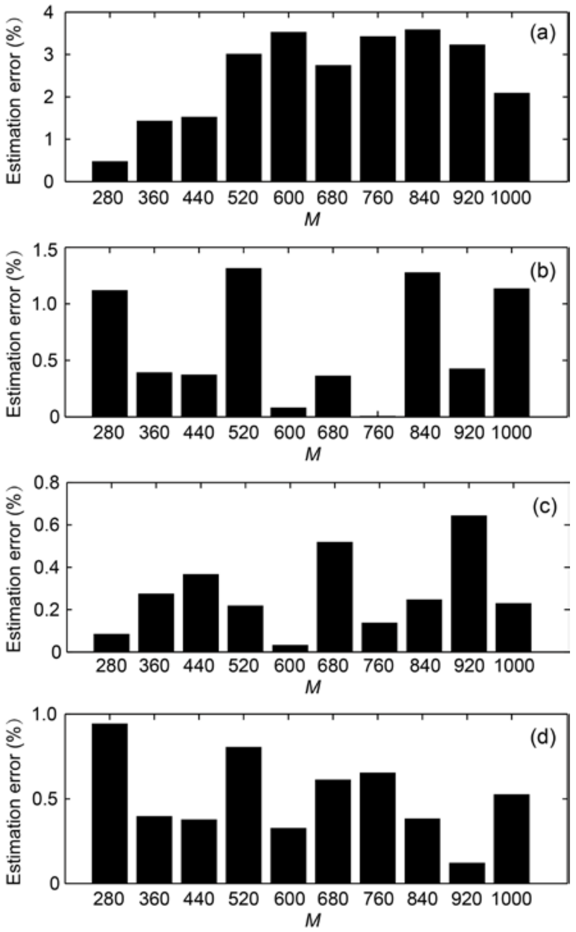

Fig. 8 Estimation errors of $\hat{f}_{first}$: (a) r=20; (b) r=25; (c) r=75; (d) r=100

4.1 Block collection completion time

Let $t_{complete}$ be the block collection completion time. $t_{complete}$ is affected by the decode delay (f_{total}) and the average length of the steps. Let τ_s denote the average length of the steps. f_{total} decreases with r, but τ_s is on the contrary because the encoding complexity increases with r, so there must be a sweet spot for r, which minimizes $t_{complete}$. τ_s consists of two parts:

1. Broadcast node selection. In each step, every node selects a random back-off time, and sets a timer according to the back-off time. The timer is decreased by 1 for every δ_s (in seconds). The first node with the timer decreasing to 0 sends a tag message to inform its neighbors that there is a broadcast node in this step. Without considering channel conflicts, we assume that broadcast nodes can be selected within δ_s. Because the tag message carries little data, its length

approximately equals the head of the wireless package, l_{ht}. Then the transmission delay of the tag message is $l_{\mathrm{ht}}/\upsilon_{\mathrm{bro}}$, where υ_{bro} is the broadcast rate. Let δ_{pd} be the propagation delay. Altogether, the delay for broadcast node selection is $l_{\mathrm{ht}}/\upsilon_{\mathrm{bro}}+\delta_{\mathrm{s}}+\delta_{\mathrm{pd}}$.

2. Broadcasting. Each selected node should produce an encoded block before it broadcasts. According to the encoding process, the maximum number of calculations for encoding is rB, and the expected amount of time used is $rB\tau_{\mathrm{c}}/2$, where τ_{c} is the average time consumed by each calculation. Because the coding vector should be broadcast along with the encoded block, the length of the broadcasted message is $l_{\mathrm{ht}}+B+r$. Its end-to-end delay is $(l_{\mathrm{ht}}+B+r)/\upsilon_{\mathrm{bro}}+\delta_{\mathrm{pd}}$ (we assume that $l_{\mathrm{ht}}+B+r<2300$ bytes), and the total delay in the broadcast is $rB\tau_{\mathrm{c}}/2+(l_{\mathrm{ht}}+B+r)/\upsilon_{\mathrm{bro}}+\delta_{\mathrm{pd}}$.

Altogether, the expected length of a step is obtained as $\tau_{\mathrm{s}}=t_{\mathrm{SIFS}}+l_{\mathrm{ht}}/\upsilon_{\mathrm{bro}}+\delta_{\mathrm{s}}+\delta_{\mathrm{pd}}+rB\tau_{\mathrm{c}}/2+(l_{\mathrm{ht}}+B+r)/\upsilon_{\mathrm{bro}}+\delta_{\mathrm{pd}}$, where t_{SIFS} is the time required for wireless devices to change from reception modes to sending modes. Combined with the fitting result of f_{total}, the block collection completion time can be calculated as

$$t_{\mathrm{complete}} \approx \hat{f}_{\mathrm{total}}\tau_{\mathrm{s}}. \quad (21)$$

Differentiating t_{complete} with respect to r, we obtain a zero point of the derivative as r^*:

$$r^* = \sqrt{\frac{Sc_{12}(2l_{\mathrm{ht}}+B)\left(\dfrac{1}{\upsilon_{\mathrm{bro}}}+\dfrac{\delta_{\mathrm{w}}}{2l_{\mathrm{ht}}+B}\right)}{(Sc_{11}+c_2)\left(\dfrac{1}{\upsilon_{\mathrm{bro}}}+\dfrac{B\tau_{\mathrm{c}}}{2}\right)}}, \quad (22)$$

where $\delta_{\mathrm{w}}=t_{\mathrm{SIFS}}+\delta_{\mathrm{s}}+2\delta_{\mathrm{pd}}$. With the increase in r, t_{complete} first decreases, and then increases. The minimum of t_{complete} is obtained at r^*.

We run the simulation using Network Simulator 2 (NS-2). Based on the data released by Wang and Li (2006) and combined with Moore's law, τ_{c} is set as 1.1618×10^{-9} s. The other parameters are listed in Table 1, where the parameters related to wireless networks are set according to Ma and Chen (2007).

The simulation results are presented in Fig. 9. The changing situation of t_{complete} is consistent with theoretical analysis. t_{complete} is a decreasing function of υ_{bro}.

Table 1 Parameter settings

Parameter	Meaning	Value
δ_{s}	The unit of back-off time	16 μs
t_{SIFS}	The time required for wireless devices to change from reception modes to sending modes	32 μs
p_{rec}	The probability that the node is not selected as a broadcast node and receives an encoded block successfully	0.5
l_{ht}	The length of the head of the wireless package	34 bytes
δ_{pd}	The propagation delay	1 μs
B	The size of each chunk	1024 bytes
S	The number of blocks of the target file	10000

Fig. 10 presents the effect of r on t_{complete} under different p_{rec}. t_{complete} decreases with the increase of p_{rec}. p_{rec} correlates positively with the channel reliability. The less reliable the channel is, the longer time it takes to complete the block collection process.

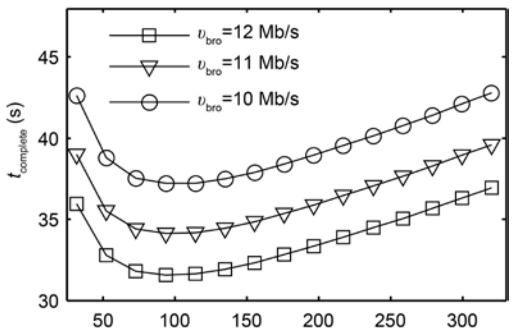

Fig. 9 Effect of r on t_{complete} with three different values of υ_{bro}

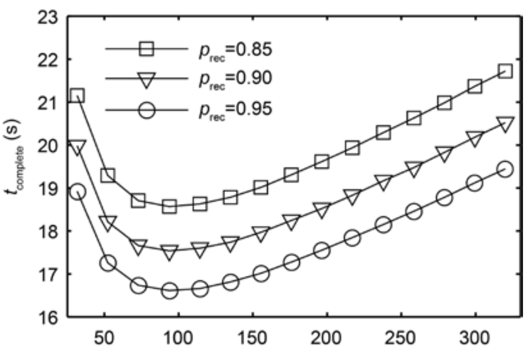

Fig. 10 Effect of r on t_{complete} with three different values of p_{rec} (υ_{bro}=12 Mb/s)

Fig. 10 also shows how t_{complete} gets its minimum value at the same r under different p_{rec}. This phenomenon can be explained using Eq. (22), where the calculation of r^* is irrelevant to p_{rec}.

4.2 Decode completion time

Let f_{first} be the decode completion time. Chunks are independent of each other, so any chunk that has collected enough blocks can be decoded. Fig. 11 shows an instance of a chunk collecting and decoding process with r=8 and M=4. The first one collecting enough blocks is chunk 2, and chunks 4, 3, 1 follow chunk 2. Chunk 2 is decoded first, and chunks 4, 3, 1 are decoded after chunk 2. Therefore, the decoding process can be modeled using a queue as shown in Fig. 11.

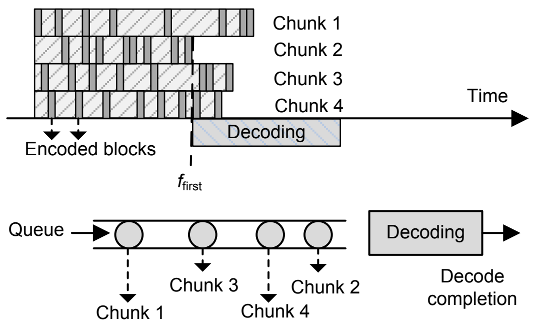

Fig. 11 An example of the decoding process

The moment when a chunk collects enough blocks corresponds to the 'arrival at the queue'. The process for when a chunk is being decoded corresponds to the 'service of the queue'. In the decoding process, chunks are decoded one by one. The decoding calculations include the inversion of the matrix (consisting of coding vectors), and the multiplication between the inversed matrix and encoded blocks. The commonly used method for matrix inversion is the Gaussian elimination, and the number of calculations for the inversion is r^3. The number of calculations for the multiplication is r^2B, so the amount of time for decoding a chunk is $h_{\text{serve}}=(r^2B+r^3)\tau_c$. The service time for the queue has a deterministic distribution with h_{serve}.

$\tau_s f_{\text{first}}$ is the time when decoding begins, after which chunks arrive at the queue one by one (Fig. 11). According to our simulation results, the arriving intervals follow the exponential distribution with an average interval of $h_{\text{arrive}}=\tau_s(f_{\text{total}}-f_{\text{first}})/M$.

Altogether, the $M/G/1/\infty$ queue can be applied. The arrival rate is $1/h_{\text{arrive}}$, and the service rate is $1/h_{\text{serve}}$. Let $N(t)$ be the length of the queue at t, where $\tau_s f_{\text{first}}$ is the time of origin, and $N(0)$=1. Denote the number of chunks that have left the queue in (0, t) as $m(t)$. Then $m(t)$ can be calculated as follows (Tang and Tang, 2006):

$$m(t) \approx \begin{cases} t / h_{\text{serve}}, & h_{\text{arrive}} < h_{\text{serve}}, \\ t / h_{\text{arrive}}, & h_{\text{arrive}} \le h_{\text{serve}}. \end{cases} \tag{23}$$

The decode completion time is the time when M chunks left, so t_{decode} can be obtained as

$$t_{\text{decode}} \approx \begin{cases} \tau_s \hat{f}_{\text{first}} + h_{\text{serve}} M, & h_{\text{arrive}} < h_{\text{serve}}, \\ \tau_s \hat{f}_{\text{first}} + h_{\text{arrive}} M, & h_{\text{arrive}} \ge h_{\text{serve}}. \end{cases} \tag{24}$$

Let r_1^* be the value of r when h_{arrive} equals h_{serve}. If $r<r_1^*$, then $h_{\text{arrive}}>h_{\text{serve}}$, and the formula for t_{decode} is the same as t_{complete}. When $r \ge r_1^*$, t_{decode} is an increasing function of r. Therefore, there should be an r that minimizes t_{decode}. If $r_1^*>r^*$, the minimum t_{decode} can be obtained at r^*; else, at r_1^*.

Fig. 12 shows the NS-2 simulation results for t_{decode}. υ_{bro} is set as 12 Mb/s. The other parameters are listed in Table 1. The change of t_{decode} agrees with the theoretical analysis. The non-solid markers indicate $h_{\text{arrive}}>h_{\text{serve}}$, i.e., $r<r_1^*$. t_{decode} first decreases, and then gets its minimum value at r^*, which is about 100 according to Fig. 12. The solid markers stand for $h_{\text{arrive}}<h_{\text{serve}}$. r_1^* is about 240 when p_{rec}=0.95, and it is 275 when p_{rec}=0.8. r_1^* is the zero point of $h_{\text{arrive}}-h_{\text{serve}}$. $h_{\text{arrive}}-h_{\text{serve}}$ decreases with p_{rec} and r, so r_1^* decreases with p_{rec}. Furthermore, we can keep increasing p_{rec} and make it 1.0 (the highest channel reliability), and the r_1^* obtained is 220. It is obvious that r_1^* is still much smaller than r^*=100, and the optimal chunk size is still min{r^*, r_1^*}=100. Therefore, under the current setting in Table 1, decreasing the channel reliability can increase t_{decode}, but it is irrelevant to the optimal chunk size. However, r_1^* may be smaller than r^* under some circumstances. In the case of $r_1^*<r^*$, the optimum chunk size is r_1^*, and the optimal chunk size will decrease with the increase of channel reliability, because r_1^* decreases with the increase of p_{rec}, and p_{rec} correlates positively with the channel reliability.

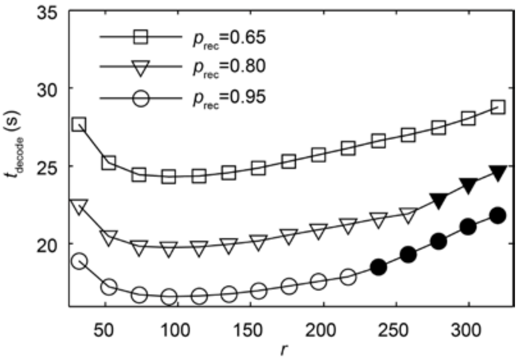

Fig. 12 Effect of r on t_{decode} with three different values of p_{rec} (v_{bro}=12 Mb/s)

5 Comparisons with existing work

There are some studies on the performance modeling of chunked NC. However, to our knowledge, we are the first to model and optimize chunked NC in cooperative downloading environments. The most related research includes Maymounkov *et al.* (2006) and Li Y *et al.* (2011), but both of them focus on centralized broadcasting systems. Maymounkov *et al.* (2006) derived the time complexity of communications in chunked NC, but the exact formula is not provided. Li Y *et al.* (2011) model chunked NC based on the coupon collector's brotherhood problem as follows:

$$f_{\text{total}}^{\text{yao}}=\int_0^\infty\left(1-\prod_{i=1}^{n}\left(1-\mathrm{e}^{-\rho_i x}E_{M_i}[S_{M_i}(\rho_i x)]\right)\right)\mathrm{d}x, \quad (25)$$

$$S_m=1+\frac{x}{1!}+\frac{x}{2!}+\cdots+\frac{x^{m-1}}{(m-1)!}, \quad (26)$$

$$E_n[m]=n\int_0^\infty\left(1-\left(1-S_m(x)\mathrm{e}^{-x}\right)^n\right)\mathrm{d}x. \quad (27)$$

The model is too complex because the integral is involved, and it is hard to make further optimizations based on the model. Therefore, Li Y *et al.* (2011) derived only a range of r, which makes the number of communications increase more slowly with M, by numerical simulation with S=1000. Compared with Li Y *et al.* (2011), we not only derive a more concise formula, but also propose calculation methods for r to optimize the system performance.

6 Conclusions

We focus on the block sharing process in cooperative downloading systems in this study. First, a system analysis model is proposed based on a Markov process, and the effect of q on the efficiency of the block collection process is derived (taking into consideration the distributed features of wireless cooperative downloading). Furthermore, based on the model and queuing theory, optimal chunk-size calculation methods are proposed to optimize the block collection completion time and the decode completion time. Compared with existing studies, we consider more details in the cooperative downloading process, including the encoding, collecting, and decoding processes, and have achieved further performance optimization. Numerical simulation shows that the block collection completion time and the decode completion time can be largely reduced using our optimal-chunk-size calculation methods, and the service experienced by users can be greatly improved. We will study chunked NC based cooperative downloading in extreme loss environments in the future.

References

Abdelrahman, O.H., Gelenbe, E., 2009. Approximate analysis of a round robin scheduling scheme for network coding. European Performance Engineering Workshop, p.212-217. https://doi.org/10.1007/978-3-642-02924-0_17

Ahmed, S., Kanhere, S.S., 2006. VANETCODE: network coding to enhance cooperative downloading in vehicular ad-hoc networks. Int. Conf. on Wireless Communications and Mobile Computing, p.527-532. https://doi.org/10.1145/1143549.1143654

Choi, J.M., So, J., Ko, Y.B., 2005. Numerical analysis of IEEE 802.11 broadcast scheme in multi-hop wireless ad hoc networks. ICOIN 2005: Information Networking, Convergence in Broadband and Mobile Networking, p.1-10. https://doi.org/10.1007/978-3-540-30582-8_1

Chou, P.A., Wu, Y., Jain, K., 2003. Practical network coding. Allerton Conf. on Communication, Control and Computing, p.40-49.

Eryilmaz, A., Ozdaglar, A., Medard, M., *et al.*, 2008. On the delay and throughput gains of coding in unreliable networks. *IEEE Trans. Inform. Theory*, **54**(12):5511-5524. https://doi.org/10.1109/TIT.2008.2006454

Heidarzadeh, A., Banihashemi, A.H., 2010. Overlapped chunked network coding. IEEE Information Theory Workshop on Information Theory, p.1-5. https://doi.org/10.1109/ITWKSPS.2010.5503153

Heidarzadeh, A., Banihashemi, A.H., 2012. Coding delay analysis of chunked codes over line networks. Int. Symp. on Network Coding, p.55-60. https://doi.org/10.1109/NETCOD.2012.6261884

Heide, J., Pedersen, M.V., Fitzek, F.H.P., *et al.*, 2009. Network coding for mobile devices—systematic binary random rateless codes. IEEE Int. Conf. on Communications Workshops, p.1-6. https://doi.org/10.1109/ICCW.2009.5208076

Joshi, G., Soljanin, E., 2013. Round-robin overlapping generations coding for fast content download. IEEE Int. Symp. on Information Theory Proc., p.2740-2744. https://doi.org/10.1109/ISIT.2013.6620725

Lee, U., Park, J.S., Yeh, J., *et al.*, 2006. CodeTorrent: content distribution using network coding in VANET. Int. Workshop on Decentralized Resource Sharing in Mobile Computing and Networking, p.1-5. https://doi.org/10.1145/1161252.1161254

Li, M., Yang, Z., Lou, W., 2011. CodeOn: cooperative popular content distribution for vehicular networks using symbol level network coding. *IEEE J. Sel. Area Commun.*, **29**(1): 223-235. https://doi.org/10.1109/JSAC.2011.110121

Li, Y., Soljanin, E., Spasojevic, P., 2011. Effects of the generation size and overlap on throughput and complexity in randomized linear network coding. *IEEE Trans. Inform. Theory*, **57**(2):1111-1123. https://doi.org/10.1109/TIT.2010.2095111

Li, Y., Vingelmann, P., Pedersen, M.V., *et al.*, 2012. Round-robin streaming with generations. Int. Symp. on Network Coding, p.143-148. https://doi.org/10.1109/NETCOD.2012.6261899

Lucani, D.E., Medard, M., Stojanovic, M., 2009a. Broadcasting in time-division duplexing: a random linear network coding approach. Workshop on Network Coding, Theory, and Applications, p.62-67. https://doi.org/10.1109/NETCOD.2009.5191395

Lucani, D.E., Medard, M., Stojanovic, M., 2009b. Random linear network coding for time-division duplexing: field size considerations. Global Telecommunications Conf., p.1-6. https://doi.org/10.1109/GLOCOM.2009.5425257

Ma, X., Chen, X., 2007. Delay and broadcast reception rates of highway safety applications in vehicular ad hoc networks. Mobile Networking for Vehicular Environments, p.85-90. https://doi.org/10.1109/MOVE.2007.4300809

Ma, X., Zhang, J., Wu, T., 2011. Reliability analysis of one-hop safety-critical broadcast services in VANETs. *IEEE Trans. Veh. Technol.*, **60**(8):3933-3946. https://doi.org/10.1109/TVT.2011.2165975

Magli, E., Wang, M., Frossard, P., *et al.*, 2013. Network coding meets multimedia: a review. *IEEE Trans. Multim.*, **15**(5):1195-1212. https://doi.org/10.1109/TMM.2013.2241415

Maymounkov, P., Harvey, N.J.A., Lun, D.S., 2006. Methods for efficient network coding. Annual Allerton Conf. on Communication, Control, and Computing, p.482-491.

Militano, L., Iera, A., Scarcello, F., 2013. A fair cooperative content-sharing service. *Comput. Netw.*, **57**(9):1955-1973. https://doi.org/10.1016/j.comnet.2013.03.014

Newman, D.J., 1960. The double dixie cup problem. *Am. Math. Mon.*, **67**(1):58-61. https://doi.org/10.2307/2308930

Pyattaev, A., Galinina, O., Andreev, S., *et al.*, 2015. Understanding practical limitations of network coding for assisted proximate communication. *IEEE J. Sel. Area Commun.*, **33**(2):156-170. https://doi.org/10.1109/JSAC.2014.2384232

Tang, Y.H., Tang, X.W., 2006. Queue Theory, Basis and Analysis Methods. Science Publishing House, Beijing, China, p.92-95 (in Chinese).

Wang, G., Lin, Z., 2014. On the performance of multi-message algebraic gossip algorithms in dynamic random geometric graphs. *IEEE Commun. Lett.*, **PP**(99):1-1. https://doi.org/10.1109/LCOMM.2014.2344047

Wang, M., Li, B., 2006. How practical is network coding? IEEE Int. Workshop on Quality of Service, p.274-278. https://doi.org/10.1109/IWQOS.2006.250480

Yu, M., Aboutorab, N., Sadeghi, P., 2014. From instantly decodable to random linear network coded broadcast. *IEEE Trans. Commun.*, **62**(11):3943-3955. https://doi.org/10.1109/TCOMM.2014.2364198

Zhang, J., Zhang, Q., Jia, W., 2007. A novel MAC protocol for cooperative downloading in vehicular networks. IEEE Global Telecommunications Conf., p.4974-4978. https://doi.org/10.1109/GLOCOM.2007.943

Zhou, H., Liu, B., Luan, T.H., *et al.*, 2014. ChainCluster: engineering a cooperative content distribution framework for highway vehicular communications. *IEEE Trans. Intell. Transp. Syst.*, **15**(6):2644-2657. https://doi.org/10.1109/TITS.2014.2321293