CoDEC: Content Distribution with (n,k) Erasure Code in MANET

Bin Dai, Wenwen Zhao, Jun Yang and Lu Lv

Department of Electronic and Information Engineering, Huazhong University of Science and Technology

ABSTRACT

The advent of smart phones has further expanded the horizon of communicative technologies. Now through the smart mobile devices, people can access and download the required data from the internet more conveniently. Many users might be interested in the same information at the same time from some hot spots. However, the respective base station may not be able to meet the demands of so many high-speed downloads at the same time, due to constraints of base station bandwidth and costs. Therefore, we present a novel content distribution system named 'CoDEC', which uses the mutual cooperation of mobile terminals by their short-range communication to implement highly efficient content distribution. The main techniques used in CoDEC are single-hop communication, (n,k) erasure code and nodes classification, which divides mobile nodes into storage nodes and normal nodes. The storage nodes can distribute coded frames, generated from the base station to the other nodes, corresponding to their requests which will save the bandwidth allocation of base station. The numerical results of simulation shows that CoDEC allows less encoding count, shorter file downloading delays compared to an existing file swarming protocol CodeTorrent.

KEYWORDS

Content Distribution, Mobile Ad hoc NETworks, Erasure Code, Node Classification.

1. Introduction

At some hot spots, simultaneous interest of many users in the download of same information might create a scenario which pushes the respective base station to its limits. For example, many tourists in a theme park, may want to download introduction video of the iconic sights via handheld devices at around the same time. However, it could be difficult for the respective base station to meet the demands of so many high-speed downloads at the same time, mostly due to base station's bandwidth constraints. In addition, the cost of downloading traffic from the base station also becomes an obstacle for users.

In this paper, we propose a novel Content Distribution system using Erasure Code (CoDEC), i.e., a file swarming protocol, which has proved to be a feasible solution for the above application scenario. In CoDEC, all mobile devices constitute a mobile ad hoc network (MANET). Initially, a part of the mobile terminals will download partial contents of the file of common interest from cellular link. Then content distribution can be boosted through the mutual cooperation of mobile terminals by their short-range communication, such as WLAN. Consequently, CoDEC could decrease the base station bandwidth allocation, the file downloading delay, energy consumption, billing, etc.

Since MANETs characterize the terminals' mobility, the network topology changes dynamically. It costs a lot of time to re-establish and maintain routing tables for each terminal [1]. Thus, the

effective usage of traditional routing protocols seems unrealistic in MANET. Many MANET P2P protocols [2] [3] try to solve this problem through cross-layer optimization. We avoid this problem by adopting single-hop communication. In single-hop communication, a mobile terminal transmits data to its neighbors which are within its physical transmission range, and its neighbors wouldn't forward the received data. Moreover, the data is propagated through overlay network of the terminals of common interest. Thus, it is not necessary to establish and maintain routing tables for the terminals, which will greatly reduce the network congestion caused by renewing the routing information.

Network coding is a notion of performing coding operations on the contents of packets throughout a network. The notion was first proposed by Ahlswede et al. [4] in 2000, who characterized the multicasting rates by showing that cut-set bounds are achievable. Where after, Li et al. [5] showed that network capacity can be achieved by using simply linear coding. The work of Li et al. was followed by Ho et al. [6] that showed that random linear combinations construct good network codes with high probability. Previous work [7] has powerfully indicated that using network coding in P2P file sharing system can effectively improve system throughput and handle dynamics of nodes in the network, including node arrival and node departure, node and piece selection, link failure, etc., thereby optimizing system performance. Therefore, we use (n, k) erasure code (a kind of network coding) in our file swarming protocol.

In previous MANET P2P file swarming protocols with network coding, peers are involved in encoding data and data dissemination [8], which have high requirements in computing, encoding, and energy of the peers. In order to alleviate the burden of terminals and reduce encoding requirement of the mobile terminals, we use (n,k) erasure code to generate all encoded blocks from the file of common interest at the base station at once, therefore avoiding intermediate node encoding again. The usage of (n,k) erasure code greatly reduces encoding count and decreases the amount of computation, which consequently makes the file downloading delay shorter. Erasure coding techniques are widely used in distributed storage systems [9] by adding data redundancy to avoid permanent data loss. Ultimately these techniques can potentially achieve orders of magnitude with higher reliability for the same redundancy compared to replication. To the best of our knowledge, our work applies for the first the erasure coding techniques to content distribution in MANET.

For the terminals involved in data dissemination, they all have high computational, coding, and storing ability, which is difficult to meet in the reality scene. Therefore, in CoDEC, cooperative mobile nodes are generally classified into two types: *storage nodes* and *normal nodes*, as shown in Figure 1. To reduce the burden and bandwidth allocation of the base station, the storage nodes obtain encoded data via exchanging information with the base station, and the normal nodes could get the required data from the storage nodes which have stored a portion of the file data. For instance, in the theme park, the mobile devices of the park staff can be chosen as storage nodes and the tourists' as normal nodes. Then we can raise overall system performance by only improving storage space, energy, etc. of the storage nodes. There is not extra capability requirements for the normal nodes. It is easy to be accepted for ordinary users in real scenarios.

Our contributions in this paper can be listed as follows:

- We propose a CoDEC protocol which is the first attempt to apply erasure coding to content distribution in MANET.
- The node classification into (n,k) erasure code based content distribution is being applied for the first time. The usage of (n,k) erasure code and node classification prodigiously reduce encoding count, decrease the amount of computation, and reduce requirements in computing and coding of normal nodes.

- To verify the CoDEC, we do a detailed simulation analysis under different simulation conditions.

The remainder of this paper is structured as follows. We give a brief introduction of the related work in Section 2. Section 3 illustrates (n,k) erasure code and node classification based file swarming protocol in detail. In Section 4, we evaluate the CoDEC system by simulation and present the simulation results. Finally, Section 5 concludes the paper.

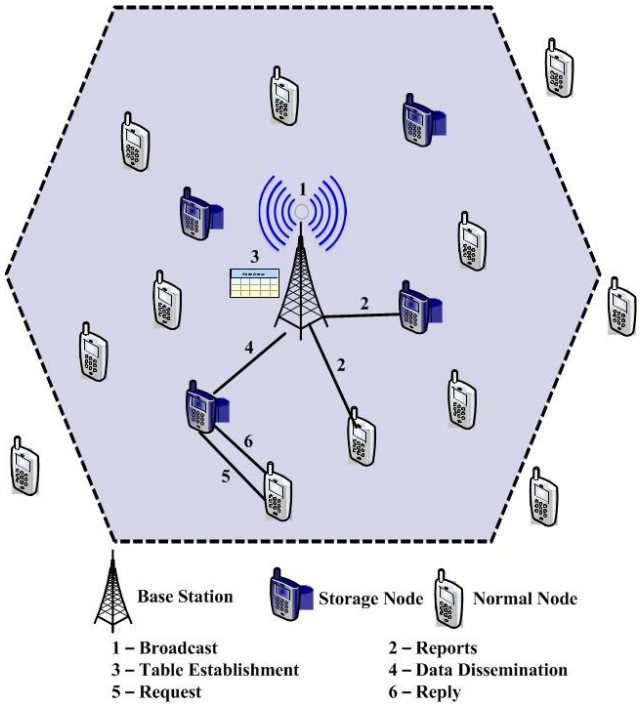

Figure 1. The content distribution system

2. RELATED WORK

In this paper, we use (n, k) erasure code to encode the file of common interest overall. Erasure codes are widely used in distributed storage systems, wireless sensor networks [10], erasure channel (BEC), and so on, by increasing the data redundancy to improve the reliability of data storage or transmission. We use lowercase boldface letters to denote constants, vectors, pieces, or frames, uppercase boldface letters to denote matrices, italics to denote variables throughout this paper.

(n, k) erasure code refers to such an encoding method: a file is divided into k pieces in a sender, i.e., $\boldsymbol{P_1}$,$\boldsymbol{P_2}$, ..., $\boldsymbol{P_k}$, each piece length b, and the sender will get n coded frames by using (n, k) erasure code to encode the k file pieces. To recovery the k original file pieces, a receiver need to acquire $q(q \geq k)$ coded frames. If the receiver gets any k coded frames, it is also able to decode the original file. Then the code is called maximum distance separable (MDS) code, or called the optimal code.

In CoDEC, the encoding algorithm is RS erasure code. RS erasure code is also the MDS code, and it is based on finite field arithmetic GF(2^w). We assume that the number of file pieces is k, and the length of pieces is b. We ultimately get n coded frames by using the formula (1).

$$\text{EncodeMatrix}_{(n,k)} \times \text{PieceMatrix}_{(k,b)} = \text{CodeMatrix}_{(n,b)} \qquad (1)$$

The selected encoding matrix EncodeMatrix has n rows, k columns ($n \geq k$) and satisfies any k rows consisting the matrixes are invertible. We use Vandermonde matrix [11] as the encoding matrix. Our Vandermonde matrix E(n, k) is constructed as formula (2), where $n \geq k$, and $i \neq j \Rightarrow x_i \neq x_j (i, j \in [1, n])$, further ($n \leq 2^w$).

$$\mathbf{E}_{(n,k)} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{k-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{k-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{k-1} \\ 1 & x_n & x_n^2 & \cdots & x_n^{k-1} \end{bmatrix} \qquad (2)$$

For ease of notation, we simplify EncodeMatrix$_{(n,k)}$, PieceMatix$_{(n,k)}$, and CodeMatrix$_{(n,b}$**Error! Reference source not found.** to $E^T = [e_1^T, ..., e_n^T], P^T = [p_1^T, ... p_k^T]$, and $C^T = [c_1^T, ... C_n^T]$ **Error! Reference source not found.**, where superscript **T** denotes the transpose operation, while c_i (i =1, …, n) is a coded frame that is a linear combination of the file pieces, **Error! Reference source not found.** , where e_{ij} is the *i*-th row, *j*-th column element of **E**, Error! Reference source not found.$_j$ (j = 1, …, k) is a file piece to be decoded, and encoding vector $e_i = [e_{i1}, e_{i2}, ..., e_{ik}]$.

CodeTorrent[8], an earlier file sharing protocol, which is also effectively suited for MANET P2P environment. In the CodeTorrent, there is a special type of node called AP, i.e. seed node, which possesses the complete file at the beginning and has intension to share the file. At first, the seed node announces the availability of the file via single-hop broadcast of the description of the file.

After receiving the file's description, if the node finds the file interesting, it will broadcast a request. Whenever requested, the seed node transmits a newly generated coded frame which is a random linear combination of the file pieces (Random Linear Network Coding, RLNC). The request of coded frames is accompanied by the null-space vector which is spanned by all encoding vectors of the frames stored in the local memory of the requesting node. On reception of such a request, a node transmits a coded frame only if there is a frame with the encoding vector in its local memory that is not orthogonal to the null-space vector received with the request. Soon after, every node can periodically broadcast the description of the file if it possesses any coded frame of the file. A node can respond to the receiving request even when it does not possess the complete file, i.e., it isn't a seed node. The reply is a newly generated coded frame that is a random linear combination of coded frames available in local memory. Until collecting enough linear independent coded frames, the node will stop broadcasting the request. Then the node can get the original file by decoding.

3. DESCRIPTION OF THE CODEC SYSTEM

In this section, we will comprehensively describe CoDEC system which is based on (n, k) erasure code and node classification. The system scenario includes a base station and some mobile nodes, as shown in Figure 1. The mobile nodes are divided into *storage nodes* and *normal nodes*. The base station covers most range of the system scenarios, and its coverage represents the area of the hot spot. The base station possesses the data to be disseminated, and it will generate coded frames by (n, k) erasure coding at the beginning. Then the coded frames will be sent out to storage nodes.

Subsequently, the storage nodes would relay these coded frames to the nodes corresponding to their requests.

3.1. (n, k) erasure coding at the base station

At the base station, a file (***F***) to be disseminated is divided into *k* pieces (**p**). By using RS erasure code, we can finally get *n* coded frames (**c**) before the process of content distribution. To recover the original file, a node needs to obtain any *k* different coded frames.

3.2. The process of content distribution

1).Base station's periodic broadcast for file description:

In MANET, all mobile nodes keep moving, at any time some may leave or enter the base station coverage, others may close wireless network interface or shutdown, etc. Therefore, the base station need to know the status of mobile nodes in time and determine how to disseminate data. In CoDEC, the base station will periodically *broadcast* the *description* of the file to the nodes within its *coverage*. The *description* contains, an identification number of a file (ID_f), file size, the number of total coded frames *n*, the essential number of coded frames to decode original file *k*, the coded frames' ID (ID_c). If the base station has multiple files to distribute, multiple descriptions are packed into the least number of packets and periodically broadcasted.

2).Report the node status at requesting nodes:

When a node receives the description of a file, if it wants to download from the base station, it will report its own *node status* that contains, for example, an identification number of node (ID_n), speeds, types (to distinguish storage nodes and normal nodes), (ID_f), the storage nodes also return stored coded frames' ID (ID_c).

3).Establishment/Maintenance of node status table:

Based on the reports of the nodes, the base station establishes and maintains *a table of node status*. When the base station receives the report of node status, it will create a list of the node status if the node report to the base station at first. Otherwise, the base station will just refresh the existing node status.

Before broadcasting the file description periodically, the base station will check all storage nodes' status to determine whether any storage node has been out of the base station coverage. If found a storage node out, the base station will randomly select a normal node to act as the storage node. The selected normal node will receive and store the coded frames that are stored in the former storage node.

4).Selection of storage nodes and data dissemination:

The base station selects some nodes from the table of node status which are in its coverage as storage nodes randomly, and it will send *n/g* coded frames to those nodes sequentially if there are *g* storage nodes in the CoDEC system. The storage nodes within the base station coverage would entirely store *n* coded frames, thus ensuring normal node could obtain *k* different coded frames to recover the original file. The encoding vector $\mathbf{e}_i$ is stored in the header of a coded frame for the purpose of later decoding [12].

After receiving a coded frame from the base station, the node will mark its own status as storage node, store the coded frame, and report its new status to the base station.

5).Periodical requests from requesting nodes:

According to the description of the file from the base station, a node that wants to acquire the file would periodically broadcast a request containing, ID_f, ID_n, ID_c of the missing coded frames, via single-hop communication.

6).Reply of coded frames from storage nodes:

The storage node within physical transmission range of the requesting node will check whether it has the requested coded frames after receiving the request. If the requested coded frames are stored, the storage node would respond to the request with the *reply*, which contains ID_n of the requesting node and the corresponding coded frames that the storage node holds. After receiving this reply, the requesting node would check whether it still needs to get these coded frames that have probably received. In case the coded frames are not yet received, or has received a part, the requesting node will store the coded frames that are still missing. Each reply is accompanied by the requesting node ID_n, so that simultaneous replies can be distinguished and matched by their ID_n.

And when the requesting node receives *k* different coded frames, it will stop sending the request, and decodes the original file pieces by using $P = E^{-1} \times C$, where E^{-1} is the inverse of encoding vector matrix **E**, which consists *k* encoding vectors from *k* different coded frames.

3.2. The process of content distribution

To further reduce the downloading delay, we adopt overhear strategy to improve the performance of CoDEC. In wireless network, a node can receive a specific packet even the node is not the designated receiver by listening to the wireless channel. In CoDEC, the nodes always overhear the packets carrying coded frames. As long as the coded frame is different from the coded frames stored in local memory, the nodes will store it. Based on overhearing, a storage node can also store some coded frames which are received from other storage nodes. Overhearing could exploit the broadcast nature of wireless medium and node mobility in full.

4. PERFORMANCE EVALUATION

In this section, we thoroughly evaluate the performance of our proposed CoDEC in different conditions, and compare with CodeTorrent, which is a previous file swarming protocol in MANET. We use OMNeT++ [13] to simulate CoDEC and CodeTorrent. However since the focus of this paper is to evaluate application layer strategies, we will keep the bottom settings agnostic. Both the CoDEC and CodeTorrent use UDP to transceiver packets without any underlying routing protocol, only relying on single-hop communication. Overhearing is also applied in the two systems. We assume that all nodes in the network are interested in downloading the same file. The simulation parameters are shown in Table 1.

Table 1. PARAMETER SETTINGS

Transport Layer Protocal	UDP
PHY/MAC Layer Protocal	CDMA2000 IEEE 802.11g
Propagation Model	Free Space Loss Model
Mobility Model	Random Waypoint mobility model
Simulation Field	600m*600m
Piece Size	4KB

We simulated a 600m x 600m field where nodes were randomly deployed. The shared file is of 1MB size and the file is only divided into a generation, the piece is 4KB, thus there are total 256 pieces *(k=256)* in the generation (as with [8]). A coded piece with encoding vector prefixed is transferred using multiple 1 KB packets. We use GF(2^8) to encode pieces, thus the size of the encoding vector is 256B which is about 6% of the piece, far less than the size of the piece. In the CoDEC, the base station is stationary and located in the center of the simulation area, and the radius of its transmission coverage is 250m obtained by controlling transmission power. The radius of the short-range link for each node is 50m in both CoDEC and CodeTorrent.

We compare the file downloading delay of CoDEC with CodeTorrent. The downloading delay is defined as the spent time that a node gets enough coded frames, i.e., *k* different coded frames in the CoDEC or *k* linear independent coded frames (the shared file is divided into *k* pieces) in the CodeTorrent, to decode the original file. For the convenience in comparison, we haven't taken the decoding time into our results as the CodeTorrent doesn't consider it either. Then we evaluate the performance of CoDEC and CodeTorrent with various configurations, for example, the number of nodes and the average speed of nodes. And we also separately analyze the impact of the *n* of (n, k) erasure code, the number of storage nodes for the downloading delay in CoDEC.

For each density condition, the experiments were repeated 50 times, and the average values were taken.

4.1. Comparison of Downloading Delay

First of all, we compared the downloading delay of CoDEC with CodeTorrent in a specific setting, which placed in a total of 30 nodes(donated the number of nodes by *N)*, i.e. *N*=30, the speed of each node(recorded as *v*) is randomly selected from 5mps to 10mps, selecting 10 nodes as storage nodes(donated the number of storage nodes by *SN*) and n=400 of (n, k) erasure code in CoDEC.
The first experimental results are shown in Figure.2. The figure clearly indicates that the CoDEC's file downloading delay reduces over 40% compared with CodeTorrent, when the proportion of completed nodes reaches 80%-90%.

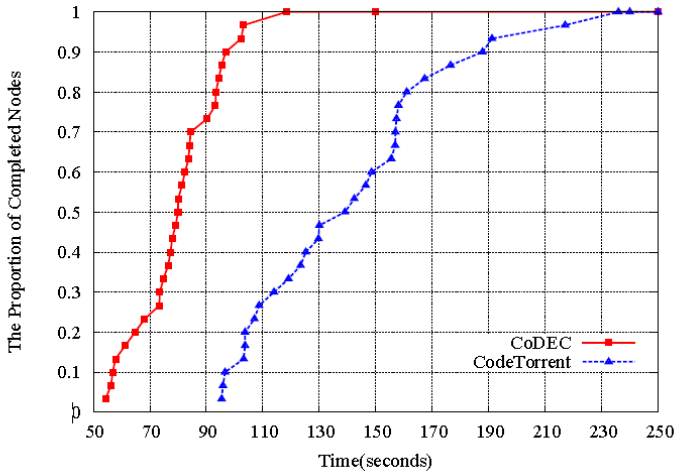

Figure 2. The proportion of completed nodes

The major reason of the improved performance is that nodes do not participate in encoding in the CoDEC. By recording the simulation progress, we learned that the average of encoding count of each node in CodeTorrent is much more than that in CoDEC, which causes a great amount of calculation resulting in the longer downloading delays, and will consume a lot of energy of mobile terminal in the real scenarios. Furthermore, a node needs to calculate the null-space vector before sending a request in CodeTorrent, which also increases the computation cost, consequently increasing the downloading delays.

4.2. Impact of *N* and *v*

Secondly, we discuss the impact of the number of nodes(*N*) on the downloading delay. *N* varies from 10 to 60, and the other settings are same as in part A. The trend of the average downloading delay with various *N* is shown in Figure.3.

Figure.3 shows that in CoDEC the downloading delay initially decreases as *N* increases, then it increases when *N* is more than 20. Because all the nodes are storage nodes at the beginning, which makes it impossible to pick up a normal node to replace the storage node that has left the coverage of the base station, thereby, part of data could be lost permanently. With the increase in *N*, the base station can find some normal nodes instead, which will ensure that all storage nodes could store more than k different coded frames in the coverage of the base station. Therefore, the downloading delay deceases. However, as *N* continually grows, the situation when multiple nodes simultaneously request to the same storage node would emerge, which will increase the collisions because of the over-occupation of the wireless channel. In addition, the probability that normal nodes encounter storage nodes also decreases. Both of the cases will increase the downloading delay. For the CodeTorrent, increase in *N*, brings more collisions on the occupied the wireless channel. Meanwhile, the engagement of more nodes, increases the number of requests received by the nodes, as overhear mechanism. Thus the average encoding count of the node also increases, which directly results in the growth of the average downloading delay, as shown in Figure.4. Meanwhile, increase in encoding count of each node goes against the expansion of the CodeTorrent.

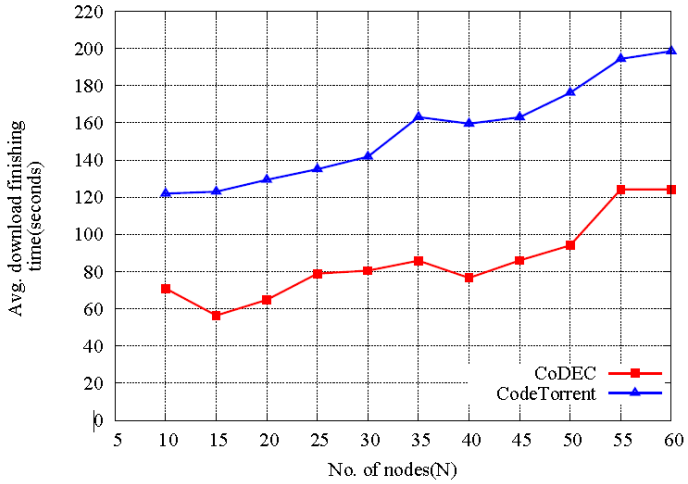

Figure 3. Impact of number of nodes on average downloading delay

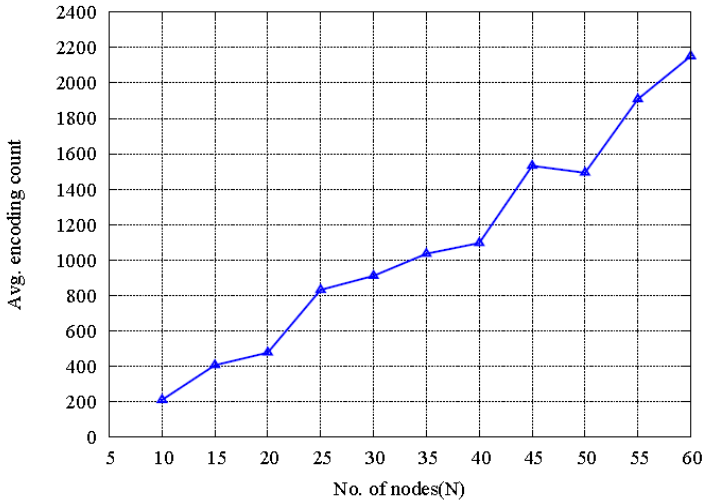

Figure 4. Impact of number of nodes on average downloading delay

Next, we investigated the impact of average speed of the nodes(v) on the average downloading delay. We changed v of the node from 1mps to 40mps. The numerical results are shown in Figure.5. In general, as the v of the node increases, for both CoDEC and CodeTorrent, the average downloading delay decreases at first, then it becomes flat. The reason is that the topology changes slowly when the nodes are moving slowly. Therefore, the node stays in prolonged contact with the same nodes and cannot get a coded frame that is different or linearly independent of the coded frames in local memory. As the v of the node increases, it becomes more probable that a node obtains helpful coded frames from the storage nodes or its neighbors. Meanwhile, the average encoding count of the node in the CodeTorrent also indicates the same trend with the downloading delay, which is shown in Figure.6.

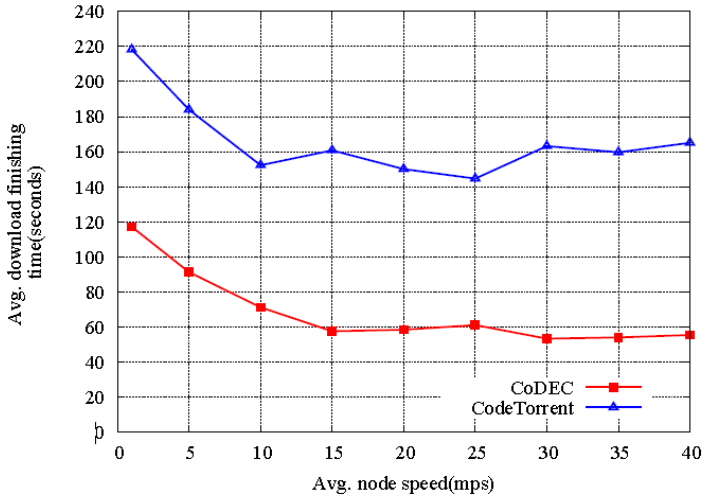

Figure 5. Impact of mobility on average downloading delay

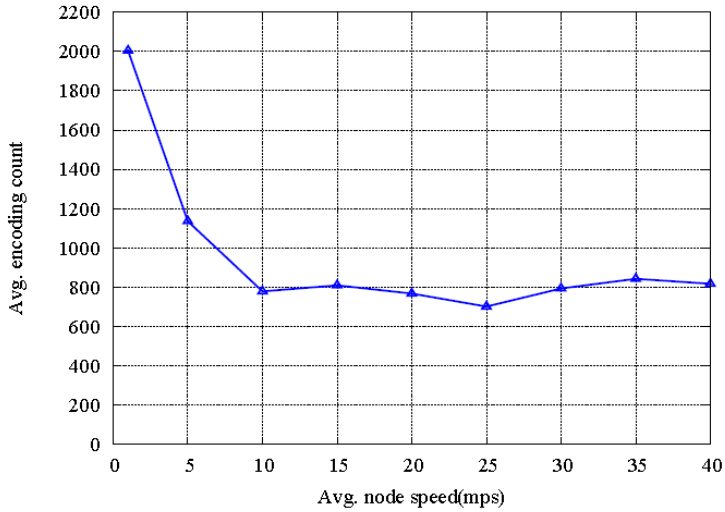

Figure 6. Impact of mobility on average encoding count in CodeTorrent

When the *v* of the node increases to a certain value, the average downloading delay flattens, mainly because of the decrease in average transmission time between the nodes, as the *v* of the node increases. It is noteworthy, if the *v* is too high, the transmission period will be too short to exchange a coded frame and this will negatively affect the performance.

4.3. Impact of *SN*, the n of (n, k) erasure code

Finally, we consider the impact of the number of storage nodes(*SN*) on the average downloading delay with a constant amount of nodes, which is 30, and the other settings are same as in part A, for the CoDEC. The result is illustrated in Figure.7.

Figure.7 clearly shows that as *SN* increases, the downloading delay drops first, then goes through a steady period, and rises at last. The probability that a node comes across the storage node is relatively low when there is little storage node in the scene, which leads to high downloading delay. As we increase *SN*, the downloading delay substantially decreases, and becomes flat. The

downloading delay begins to increase when the proportion of the *SN* exceeds a threshold, as is shown in Figure. 7, due to the excessive storage nodes, resulting in the coded frames are excessively scattered.

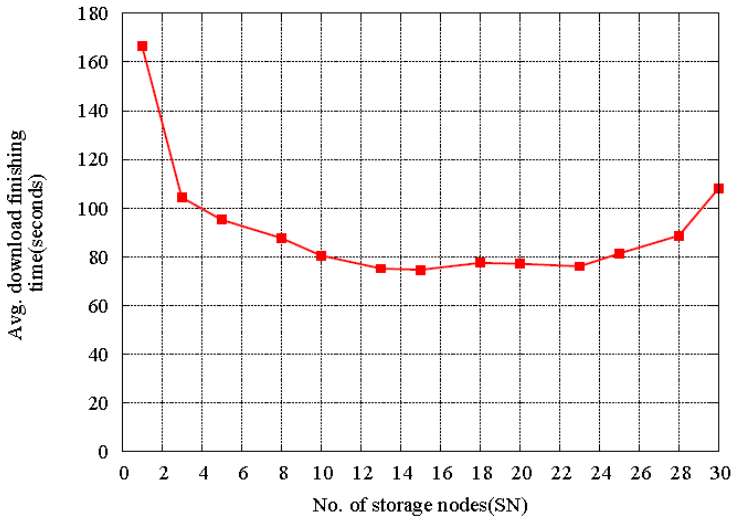

Figure 7. Impact of the number of storage nodes on average downloading delay in CoDEC

Next, we surveyed the impact of the *n* of (n, k) erasure code in the case of various number of storage nodes (s = 3,10,20,28) for the CoDEC. We changed *n* from 260 to 500, and kept *k* always at 256. The simulation results are shown in Fig.8. As the *n* grows, the whole downloading delay decreases first, then changes to be steady. This is due to the redundancy of coded frames increases in the CoDEC, and the number of stored coded frames in each storage node increases as well, so a node can obtain more different coded frames by interacting with a storage node. Owing to limited transmission time between nodes, a node cannot successfully get a large number of coded frames at once. Thus the downloading delay becomes flat as *n* further grows. From the Figure.8, we also obtain that the downloading delay is relatively large and unstable when the number of storage nodes is less. And when the number of storage nodes over a threshold, the curves of downloading delay almost overlap.

From the numerical results, we can learn that selecting the appropriate number of storage nodes and the adaptive *n* is crucial to the system performance.

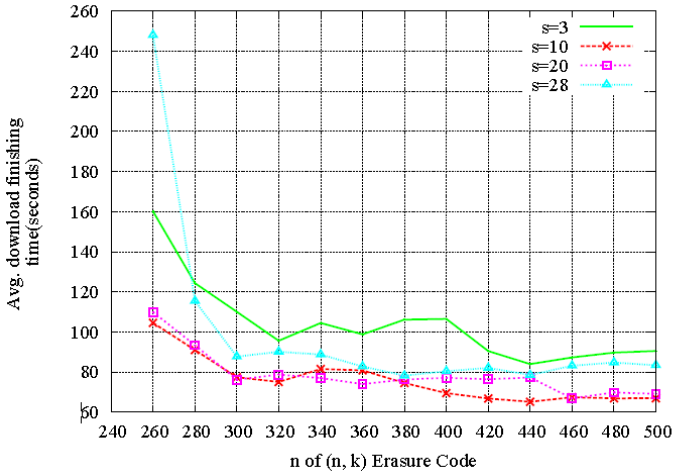

Figure 8. Impact of the n of (n, k) erasure code on the average downloading delay in CoDEC

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel content distribution system CoDEC, i.e., a file swarming protocol, based on (n, k) erasure code and node classification. Via simulation, we show that our protocol saves about 45% of the file downloading delay compared to CodeTorrent. In addition, CoDEC system saves the bandwidth allocation of the base station and reduces requirements in computing, coding, and energy of nodes, which is also closer to the needs of users in the real scenarios.

In the future research, we will focus more thoroughly on the influence of node mobility in a real environment where the node mobility patterns are more complex than we have considered in this paper.

6. ACKNOWLEDGEMENTS

This work was supported by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant no. 2012BAH93F01, the Innovation Research Fund of Huazhong University of Science and Technology, no. 2014TS095 and the National Science Foundation of China under Grant no. 60803005.

REFERENCES

[1] A. Nandan, S. Das, G. Pau, M. Sanadidi, and M. Gerla, “Cooperative downloading in Vehicular Ad Hoc Networks”, In 2nd Annual Conference on Wireless On demand Network Systems and Services (WONS), 2005.

[2] Jun-Li Kuo, Chen-Hua Shih, and Yaw-Chung Chen, “A Cross-Layer Design for P2P Live Streaming with Graceful Handover in Mobile IP Network”, In the 13th International Conference Intelligent Transportation Systems Telecommunications (ITST), Nov. 2013.

[3] Evariste Logota, Hugo Marques, and Jonathan Rodriguez, “A Cross-layer Resource Over-Provisioning Architecture for P2P Networks”, In 18th International Conference on Digital Signal Processing (DSP), July 2013.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow”, IEEE Trans. Inform. Theory, 46(4):1204-1216, July 2000.

[5] Li S-Y R, Yeung R W, and Cai N, "Linear network coding", IEEE Trans Info Theory, 49(2):371-381, 2003.
[6] Ho T, Medard M, Shi J, et al, "On Randomized Network Coding", In: 41st Annual Allerton Conference on Communication Control and Computing, Oct. 2003.
[7] Dinh Nguyen and Hidenori Nakazato, "Centrality-Based Network Coder Placement for Peer-To-Peer Content Distribution", International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.3, May 2013.
[8] Lee, U., Park, J., Lee, S., and Ro, W, "Effcient Peer-to-peer File Sharing using Network Coding in MANET",Journal of Communications and Networks (JCN), Special Issue on Network Coding, vol.10, pp.422-429, 2008.
[9] Hu, Y., Yu, C., and Li, Y, "NCFS: On the Practicality and Extensibility of a Network-Coding-Based Distributed File System", Network Coding (NetCod), pp.25-27, July 2011.
[10] Alawadhi, R. and Nair, S., "A joint scheme for secure and reliable communication in wireless sensor networks, In ACS International Conference on Computer Systems and Applications", AICCSA 2013, May 2013.
[11] James S.Plank, "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems", Software-Practice \& Experience, 27(9):995-1012, September 1997.
[12] P. Chou, Y. Wu, and K. Jain, Practical network coding, In Proc. Allerton, 2003.
[13] Network Simulation Framework, http://www.omnetpp.org.
[14] Tracy Camp, Jeff Boleng, and Vanessa Davies, "A Survey of Mobility Models for Ad Hoc Network Research", Wireless Communications and Mobile Computing, pp.483-502, 2002.

Authors

Bin Dai received the B. Eng, the M. Eng degrees and the PhD degree from Huazhong University of Science and Technology, P. R. China in 2000, 2002 and 2006, respectively. From 2007 to 2008, he was a Research Fellow at the City University of Hong Kong. He is currently an associate professor at Department of Electronic and Information Engineering, Huazhong University of Science and Technology, P. R. China. His research interests include wireless network, network coding, software-defined network.

WenWen Zhao received the Master degree from Huazhong University of Science and Technology.Her research interests include wireless network, network coding, software-defined network.

Jun Yang received the PhD degree from Huazhong University of Science and Technology of China, P. R. China in 2013. He is currently a Postdoctoral Fellow in Huazhong University of Science and Technology. His research interests include network coding and software-defined network.

Lu lv received the bachelor's degree from HuBei University.she is currently a master student in Huazhong University of Science and Technology.Her research interests include network coding, software-defined network.