J. Basic. Appl. Sci. Res., 2(7)6588-6595, 2012
© 2012, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied
Scientific Research
www.textroad.com

The Efficiencies of Decoding Methods in Random Network Coding for Video Streaming

Behrang Barekatain[1], Mohd Aizaini Maarof[1], Hamid Reza Ghaeini[2]

[1]Faculty of Computer Science & Information System Universiti Teknologi Malaysia, Johor Bahru, Malaysia
[2]Faculty of Electrical & Computer Engineering, Tarbiat Modares University, Tehran, Iran

ABSTRACT

In the recent years, video streaming over wireless networks has become very popular. Users can join the network easily and move from one point to another with no restriction. On the other hand, there are some problems in providing smooth video playback in these nodes due to time-varying channels, obstacles and low upload and download bandwidth, especially in gadgets such as mobile phones. Although it is possible to degrade the side effects of these problems by using some efficient video compression techniques, better and more efficient solutions are required to cope with them. Random Network Coding (RNC) promises high video quality in receivers by increasing encoded packets diversity. However, decoding these packets is a challengeable subject, because of imposed delay to the system. This study compares two existent techniques in decoding in terms of computation time. The results show that the efficiencies of these methods depend on both the number and the size of the blocks. Moreover, the Gauss-Jordan elimination method provides better efficiency when the number of video blocks exceeds 256.

KEYWORDS: Random Network Coding, Block Size, Video.

I. INTRODUCTION

Multimedia dissemination will be one of the most expensive and popular applications in near future. Video-on-Demand (VoD) and live video streaming are two main classifications of video dissemination [1, 2]. Providing smooth video playback is difficult in wireless networks due to existent interferences, noises and obstacles between transceiver and receiver as well as the mobility of nodes. Providing in demand level of QoS (Quality-of-Service) is one of the most challengeable issues in time-varying channels. Variable-Bit-Rate (VBR) channels, contrary to Constant-Bit-Rate (CBR) channels, can be used in networks with time-varying channels to keep required level of QoS in the receivers [3].

In wireless networks, it is not useful to encode the video frames with same bit stream, because the conditions of channels, heterogeneous bandwidths and the position of nodes may be changed from time to time [4]. VoD and live video streaming are different in technical aspects. Synchronized video playback on all nodes is the main characteristic of live streaming, while in VoD users can join the system whenever they want and watch the video from the first frame. In other words, playback of the same video stream on different nodes is not synchronized. This leads to more flexibility in video-on-demand streaming. YouTube and PPLive are two famous VoD-based and live-based systems, respectively. One of the most important techniques to cope with delay constraints, especially in live streaming, is buffering. All applications ask their users to buffer few seconds of the video stream before starting the playback. However, buffering is not enough in order to have smooth video playback, because users just tolerate the initial buffer stage and it is not acceptable for them to pause the playback due to buffering process. Video compression and efficient packet delivery are two important approaches to help users to enjoy high video quality on their gadgets. Among all existent video compression methods, the H.264/AVC standard introduces many efficient methods for video compression [5, 6]. Each video segment contains at least two GoPs (Group-of-Pictures) and each GoP includes some frames. These frames are Intra-frame (I), Predictive-inter-frame (P) and Bi-predictive-frame (B). Receivers can decode I-frames independently; while decoding a P-frame depends on the previous I-frame and P-frame. Decoding a B-frame depends on both previous and next I-frames and P-frames. Different ordering of these frames can be considered in a GoP based on the used profile in the standard. Fig. 1 depicts a GoP including 9 frames and their dependencies.

The H.264/SVC standard [6], an extension of the H.264/AVC, introduces three scalability methods including temporal, spatial and quality to provide better video quality in nodes with heterogeneous bandwidth. Temporal scalability method encodes video frames in different frame rate, while in spatial scalability the encoder encodes frames with different spatial resolution. In the third method, quality scalability, each sub-stream has same spatial-temporal resolutions, but with lower SNR (Signal-to-Noise-Ratio). The SVC encoder sends the video frames with the lowest quality as base-layer and enhanced-layers includes same frames but with higher quality. As a result, each node which gets the base-layer starts the video playback and better video quality will be achieved if it can download more enhanced-layers. However, it is necessary to consider dependency among these layers. For example, it is impossible to decode a P-frame in the enhanced-layer 1 if the correspondence P-frame in the base-layer has not

---

***Corresponding Author:** Behrang Barekatain, Faculty of Computer Science & Information System, Universiti Teknologi Malaysia, Johor Bahru, Malaysia received yet. Many recent works such as [7] used SVC for video streaming with network coding. In addition to using an efficient video compression method and according to mentioned recent works, network coding [8] increases video quality among clients in wireless mesh networks (WMNs) [9-12] by addressing mentioned problems . The efficiency of using network coding is examined in many works such as [13, 14].

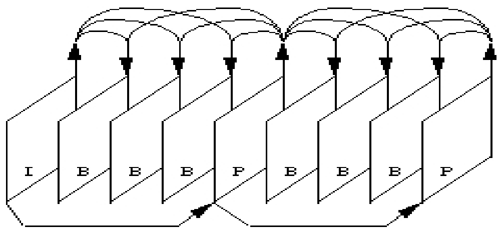


FIGURE 1: A GoP with 9 frames.

The rest of this paper is organized as follow. Different network coding methods are discussed in section II. Section III defines the problem statement of the research to show the effects of different block sizes on decoding delay and the simulation results of comparing different decoding approaches are presented in section IV. Finally, the paper is concluded in section V.

II. NETWORK CODING

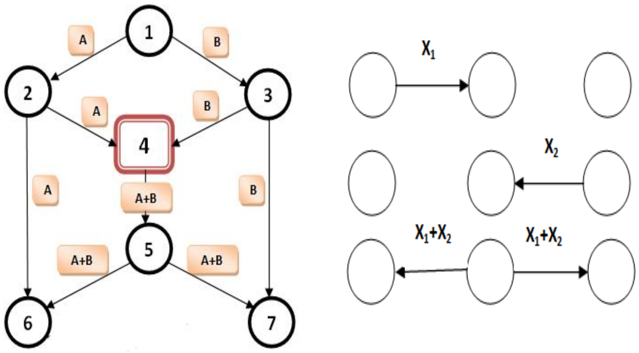
A. Definition

Network Coding (NC) was first introduced in 1999 by R. W. Yeung and Z. Zhang as an alternative to routing. Serious works on using network coding began from 2003. Video frames can be divided into many small packets, each of them may route from different path to the destination. As a result, an intermediate node can combine received packets and apply simple XoR operation on them. This is shown in Fig. 2(a). In this figure, the source node 1 delivers two bits A and B to nodes 2 and 3, respectively. Node 4 applies XoR operation on both bits A and B and sends one bit A+B instead of two bits.

This not only reduces the number of transmission, but also lets nodes 6 and 7 extract required bits B and A sooner than usual by applying XoR operation on (A , A+B) and (B , A+B), respectively. In fact, end-to-end delay between source and destinations decreases which leads to higher video quality, because receivers can have their required video frames earlier than usual. Moreover, according to Fig. 2(b), network coding decreases energy consumption in wireless nodes which is very important, especially for small gadgets such as mobile phones.

B. Network Coding Approaches

Network coding approaches provides both some advantages and some drawbacks. XoR-based, as illustrated in Fig. 2, is the simplest approach which applies XoR operation on the received video segments or blocks. By using network coding, a video frame divides into some segments and each segment future is divided into many *k*-byte blocks. Usually, the sizes of segments and blocks within a frame and segment are the same, respectively.



(a) (b)

Figure 2: Network coding efficiency

XoR-based approach is topology dependent. Random Network Coding (RNC) [15] addresses this problem. In fact, a node can apply this approach on different blocks of a segment. RNC encodes received blocks by multiplying some coefficients to the original blocks [16]. In random network coding, each node selects some random coefficient in $GF(p^m)$ and generates a coefficient matrix (C_{nn}), where GF (Galois Field) is a finite field, p is a primary and m is a positive value [17]. All mathematical operations such as multiplication are performed in GF. Then, it selects *n* original *k*-byte blocks and arranges them in the data matrix (B_{nk}).The encoded matrix X_{nk} can be achieved by using the (1). Nodes who receive the encoded matrix X can decode it by using the (2).

$$X_{nk} = C_{nn}.B_{nk} \quad (1)$$

$$B_{nk} = C^{-1}_{nn}.X_{nk} \quad (2)$$

This increases the diversity of blocks in receivers. In fact, each receiver needs to get maximum *n* encoded blocks related to a segment from different seeders to decode the segment. Each of these two network coding approaches can be accompanied with greedy or hierarchical methods. In greedy network coding, each node tries to encode those blocks which are decodable in more number of neighbors. This can be achieved by comparing the received buffer map status (BMS) messages from neighbors. Each node sends the required segments to its neighbors by generating a bitmap message to show which segments will be requested in near future or which segments are stored in its buffer. In hierarchical network coding, the encoder gives higher priority to the frames belonging to the lower layers (e.g. the base-layer) [18]. Table 1 compares some previous works since 2010 in terms of used network coding approach. Some of them used more than one approach to achieve better performance. Raptor network coding is another method which is introduced to increase resilience in time-varying channels [19]. Raptor network coding method uses a feedback message from the destinations to the source in order to adjust the redundancy degree of the encoded blocks. Therefore, this method is not suitable for live video streaming, because it increases end-to-end delay [19].

Table 1.
Compare previous works in terms of used NC

[Ref.]	**XOR**	**RNC**	**Greedy**	**[Ref.]**	**XOR**	**RNC**	**Greedy**
[20]		√		[21]		√	
[22]	√		√	[23]	√		
[24]		√		[25]		√	
[13]		√		[26]		√	
[27]	√		√	[28]		√	

III. PROBLEM STATEMENT

As illustrated in table 1, RNC is the most recent used approach due to its ability in increasing the diversities of frames. Moreover, it eliminates the need for having a centralized knowledge about the network topology and makes the system more robustness when the network topology changes or peers join and leave the network repeatedly. However, computation complexity and transmission overhead due to sending the coefficients matrix cannot be ignored in this method, especially when the size of blocks is small. Decoding process causes a large portion of computation complexity in RNC. Gauss-Jordan Elimination and basic decoding are the two most important methods for decoding process. In Gauss-Jordan elimination method, each receiver arranges received coefficients vector C_i and encoded block X_i and apply Gauss-Jordan elimination method on them to make a diagonal matrix such that all values on the main diagonal be equal to 1 [29]. Although this method is progressive such that the decoder starts decoding as soon as the first encoded vector of matrix X is received, it imposes high computation complexity to the system. Moreover, the overhead due to sending the coefficients matrix C is considerable, which is as a drawback in WMNs where gadgets have not enough large bandwidth. In the second method, basic decoding, receivers have to wait for the whole coefficients matrix C to be received. Then, they can decode matrix B using (3). Adj gives the Adjoint of matrix C.

$$B = \left(\frac{1}{\det(C)}\right) * Adj(C) * X_{nk} \quad (3)$$

Although this method decreases computation complexity existent in the Gauss-Jordan elimination method, it is not progressive which can causes more delay in delivering required segments to the video player. These challenges are mentioned in many previous works such as [20, 21, 24, 29-33]. According to mentioned problems, it is necessary to find the best block size for encoding the original video frames. In addition, the effects of changing in the block size (*k*) and the number of existent blocks in matrix B (*n*) are other important issues in the encoding process. In the next section, we will analysis these parameters to clarify the effects of these changes in the size and the number of blocks in matrix B to find the best values for *n* and *k*.

IV. SIMULATION RESULTS

According to the statement of problem, different block sizes *k*=[64B,128B,256B,512B,1024B,2048B,4096B,8192B] and various number of blocks *n*=[64,128,256,512,768] were simulated in MATLAB R2010a on a computer with a 1.6 GHz CPU Intel core i7 and 4GB of physical memory. This simulation ran for five times and the averaged results for encoding and Gauss-Jordan elimination as well as basic decoding are depicted in figures 3 to 6. In the following, we examine these results in detail.

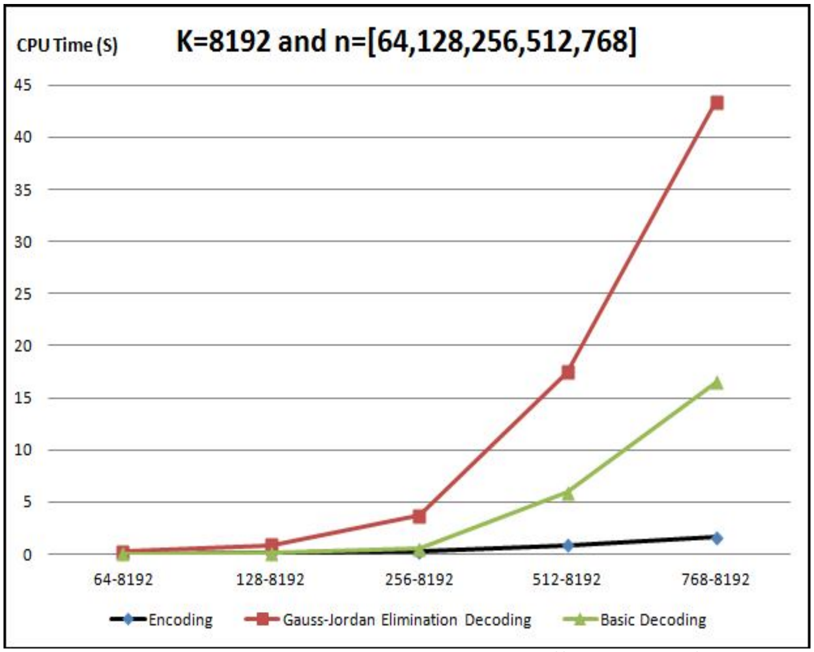
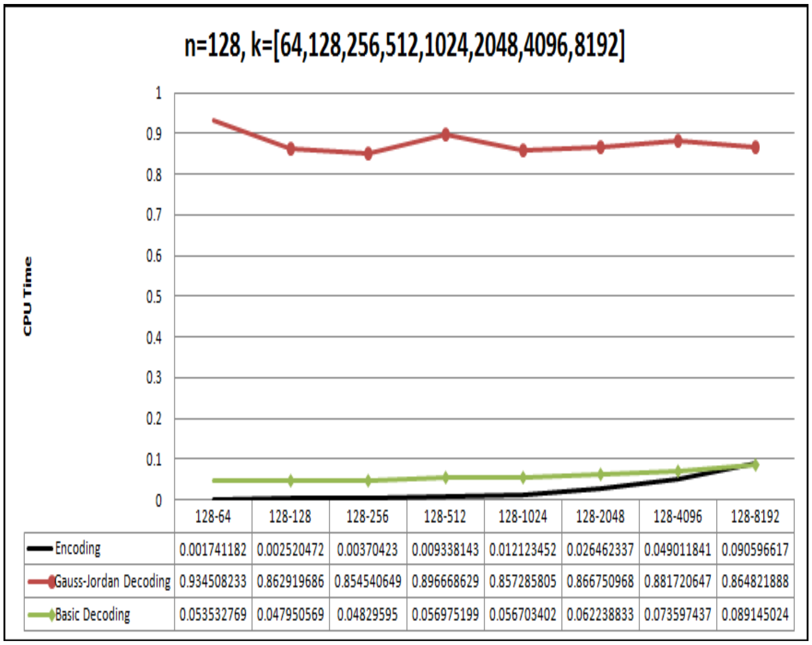


Figure 3: Compare CPU time with K=8192 and different values of n



	128-64	128-128	128-256	128-512	128-1024	128-2048	128-4096	128-8192
Encoding	0.001741182	0.002520472	0.00370423	0.009338143	0.012123452	0.026462337	0.049011841	0.090596617
Gauss-Jordan Decoding	0.934508233	0.862919686	0.854540649	0.896668629	0.857285805	0.866750968	0.881720647	0.864821888
Basic Decoding	0.053532769	0.047950569	0.04829595	0.056975199	0.056703402	0.062238833	0.073597437	0.089145024

Figure 4: Compare CPU time with n=128 and different values of k

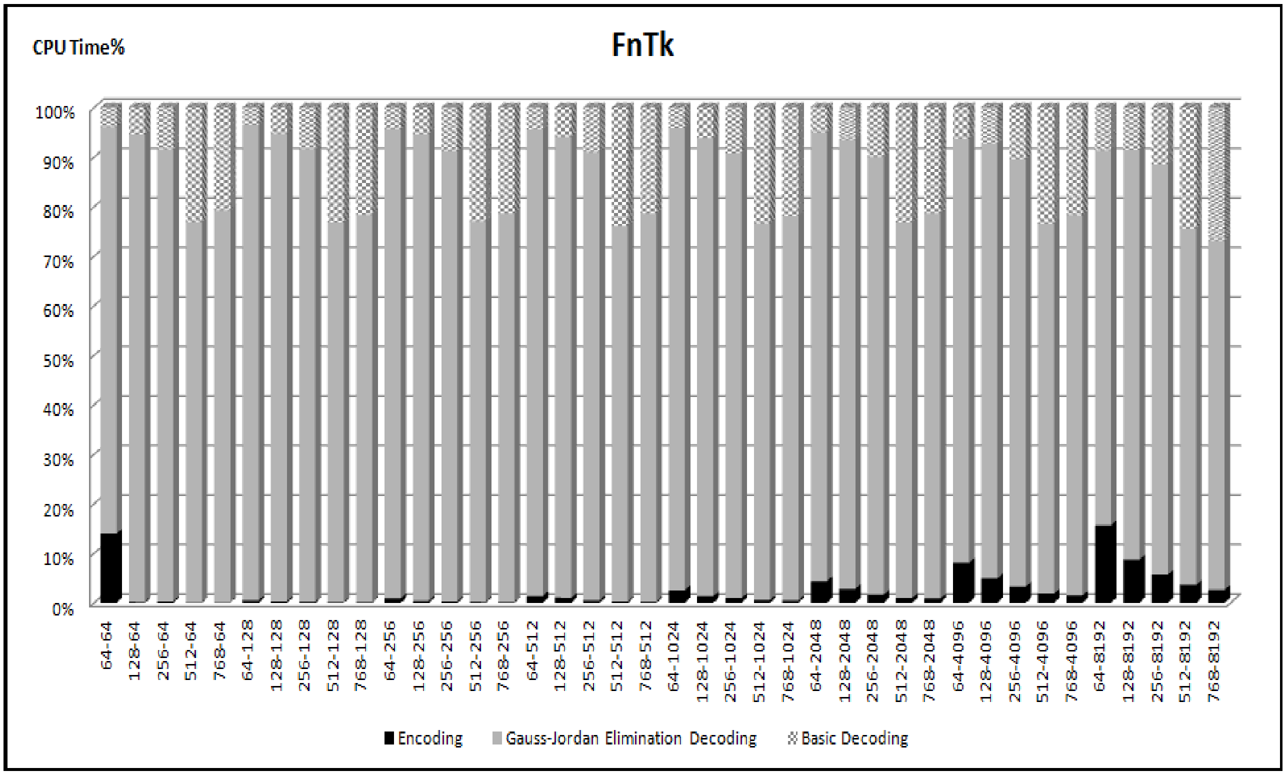


Figure 5: Compare CPU time when first the number of blocks, then the blocks size changes

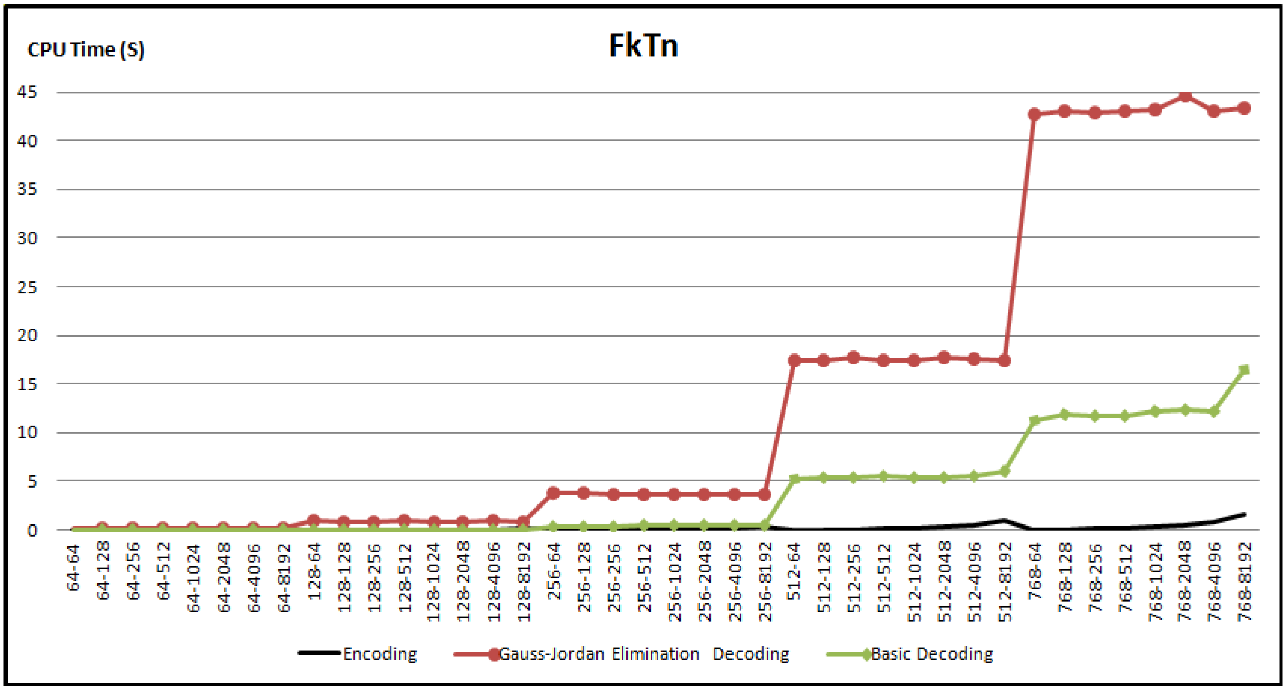


Figure 6: Compare CPU time when first the blocks size, then the number of blocks changes

In the following, these figures are explained in summary:

- Fig. 3: various amounts of *n* with *k* equals to 8192 bytes are considered in this figure to compare the effect of different number of blocks, when the size of block remains constant. We select the maximum value for *k* in this figure to show a precise comparison with large block size.
- Fig. 4: in this test, different amounts of *k* are examined when the number of blocks remains constant and equals to 128. This figure examines the effect of variable block sizes on encoding and decoding processes. We consider 128 blocks, because it is more practical due to existent computation complexity in random network coding.
- Fig. 5 and Fig. 6: these figures compare the computation complexity for all considered values of *n* and *k* to show the effects of different number of blocks and block sizes on the computation complexity.

What can be inferred from Fig. 3 is that encoding consumes low CPU time and it just increases very slightly when the number of blocks exceeds 256, while basic decoding increases moderately in this point. However, required CPU time in the Gauss-Jordan elimination method increases slightly up to n=128, then heightens moderately up to n=256 before increasing sharply up to n=768.

This is approximately same with other values of k. As a result, it is better to use basic decoding when the number of blocks is fewer than 256. On the other hand, albeit the Gauss-Jordan elimination method consumes more CPU time, it outperforms the basic decoding approach when the number of blocks exceeds 256, because it provides progressive decoding such that the decoding process starts as soon as the first encoded block is received. According to this fact that the uploading bandwidth of a gadget is 1Gbps, at least 113 seconds need to upload all encoded blocks in case of n=768 and k=8192. It means that by using progressive decoding, the decoding times are concealed within the required time for receiving the segment in most of the time.

In Fig. 4, *n* remains constant while *k* changes in the mentioned range from 64 bytes to 8192 bytes. Encoding times increases slightly with *k* greater than 1024B. The basic decoding CPU time increases very slightly, while the required CPU time in Gauss-Jordan elimination method is fluctuated with different amounts of k. It means that the required CPU time in this method does not depend on *k* and changes rapidly when the number of blocks (*n*) increases. The results show that the effect of the number of blocks on the required CPU time is more considerable than that of size of blocks in both decoding methods.

Figure 5 and 6 compare the CPU time for all considered values of *n* and k, where in Fig. 5 first the number of blocks changes for each value of *k* (FnTk[1]) and in Fig. 6 it is vice versa (FkTn[2]). Fig. 5 compares the percentage of consumed CPU time by each pair value of *n* and *k* (*n-k*) for encoding and two decoding methods. The encoding time in case of (64,64) is considerable. This means that it is better to have more than 64 blocks in encoding process. Interestingly, the required encoding time is considerable when *n* equals to 64 for different amounts of *k*. Moreover, the results of Fig. 4 are confirmed in this figure such that decoding times are approximately same for various amounts of *k* with same values of *n* (e.g. n=128).

In Fig. 6, by increasing the number of blocks (*n*), encoding time grows very slightly, while both the Gauss-Jordan elimination and the basic decoding methods jumped in the consumed CPU time when the number of blocks changes. In fact, this jumping is completely considerable, when the amount of *n* becomes greater than 256. However, as mentioned before, the Gauss-Jordan elimination method outperforms the basic method due to using progressive decoding in receivers.

All in all, we can relinquish the encoding time in random network coding. Finally, each random network coding framework or architecture needs to consider proper values for *n* and *k*. Moreover, it is better to consider proper value for *k* according to the selected number of blocks. In fact, the computation complexity of random network coding is considerable when the number of blocks is large and the size of each block is small. For example, as illustrated in Fig. 6, when the number of blocks is 768, we can see that required CPU time increases very slightly when *k* begins to increase from 64 to 8192 bytes in both Gauss-Jordan elimination and basic decoding methods. Consequently, there is no benefit to choose small block size. However, required bandwidth for sending large block sizes need to be considered, especially in wireless nodes with limited and heterogeneous bandwidths.

V. CONCLUSION

Although network coding introduces security issues on intermediate nodes, it increases the throughput and the robustness of the network, when network topology changes. Live video streaming is delay sensitive and it is necessary to have an efficient method for disseminating it. Moreover, although wirelesses mesh networks provides many benefits such as self-healing and self-organization for increasing the scalability and the stability of the system, time-varying channels and the mobility of nodes are two important challenges in this type of networks which can degrade the perceived video quality.

Network coding, especially random network coding, promises a good solution to combat these issues. However, required CPU time in encoding and decoding processes, irrespective of the transmission time, should be considered when a network coding framework decides to select the proper values for *n* and *k*. This work compares the required encoding and decoding times for both Gauss-Jordan elimination and basic decoding methods and the results show that the first method is more suitable for video

---

[1] First n changes, Then k changes

[2] First k changes, Then n changes

streaming when the number of blocks exceeds 256, because of progressive property in decoding. This leads to smooth video playback in both mobile and stationary nodes in WMNs, because the receiver can decode the encoded blocks in a short time. Finally, the effects of using different values of *n* and *k* are more considerable, when the video stream is live.

REFERENCES

[1] Ma, K.J., R. Bartoš, and S. Bhatia, *A survey of schemes for Internet-based video delivery*. Journal of Network and Computer Applications, 2011. **34**(5): p. 1572-1586.

[2] Driscoll, N., et al., *Streaming MPEG4 Video over Wired and Wireless Local Area Networks*, in *A local conference in Missouri University*. 2009.

[3] Montazeri, S., M. Fathy, and R. Berangi, *An adaptive fair-distributed scheduling algorithm to guarantee QoS for both VBR and CBR video traffics on IEEE 802.11e WLANs*. EURASIP J. Adv. Signal Process, 2008. **2008**: p. 1-12.

[4] Hiertz, G.R., et al., *IEEE 802.11s: The WLAN Mesh Standard*. Wireless Communications, IEEE, 2010. **17**(1): p. 104-111.

[5] Wiegand, T., et al., *Overview of the H.264/AVC video coding standard*. Circuits and Systems for Video Technology, IEEE Transactions on, 2003. **13**(7): p. 560-576.

[6] Richardson, I.E., ed. *The H.264 advanced video compression standard*. 2010.

[7] Razzaq, A. and A. Mehaoua. *A robust network coding scheme for SVC- based streaming over wireless mesh network*. in *Information Infrastructure Symposium, 2009. GIIS '09. Global*. 2009.

[8] Fragouli, C. and E. Soljanin, *Network Coding Fundamentals*. Vol. 2, NO. 1. 2007: Fundamentals and Trends in Networking.

[9] Ming-Xin, H. and K. Geng-Sheng. *Delay and throughput Analysis of IEEE 802.11s Networks*. in *Communications Workshops, 2008. ICC Workshops '08. IEEE International Conference on*. 2008.

[10] Zhang, Y., J. Luo, and H. Hu, *Wireless Mesh Networking: Architectures, Protocols and Standards (Wireless Networks and Mobile Communications)*. 2007: Taylor & Francis Group, LLC.

[11] Akyildiz, I.F., X. Wang, and W. Wang, *Wireless mesh networks: a survey*. Computer Networks, 2005. **47**(4): p. 445-487.

[12] Akyildiz, I.F. and W. Xudong, *A survey on wireless mesh networks*. Communications Magazine, IEEE, 2005. **43**(9): p. S23-S30.

[13] Lu, J., S. Xiao, and C. Wu, *Efficient broadcasting for scalable video coding streaming using random linear network coding*, in *SPIE, Satellite Data Compression, Communications, and Processing VI* 2010: San Diego, California, USA

[14] Ding, Y., et al., *Peer-to-peer video-on-demand with scalable video coding*. Computer Communications, 2010. **33**(14): p. 1589-1597.

[15] Fragouli, C. and E. Soljanin, *Network Coding Applications*. 2007: now published Inc.

[16] Cuiping, J., et al. *A Packet Loss Protection Scheme Joint Deterministic Network Coding and Random Linear Network Coding for H.264/AVC*. in *Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on*. 2011.

[17] MacKay, D.J.C., *Information Theory, Inference, and Learning Algorithms*. 7.2, fourth printing ed. 2005: Cambridge University Press.

[18] Jingjing, S., et al. *Layered Network Coding and Hierarchical Network Coding for Peer-to-Peer Streaming*. in *Circuits, Communications and Systems, 2009. PACCS '09. Pacific-Asia Conference on*. 2009.

[19] Thomos, N. and P. Frossard, *Raptor network video coding*, in *Proceedings of the international workshop on Workshop on mobile video*. 2007, ACM: Augsburg, Bavaria, Germany. p. 19-24.

[20] Li, Y.-S., H.-Y. Teng, and R.-H. Hwang, *P2P SVC-encoded video streaming based on network coding*, in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. 2010, ACM: Caen, France. p. 1277-1281.

[21] Anh Tuan, N., L. Baochun, and F. Eliassen. *Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding*. in *INFOCOM, 2010 Proceedings IEEE*. 2010.

[22] Seferoglu, H. and A. Markopoulou. *Delay-Optimized Network Coding for Video Streaming over Wireless Networks*. in *Communications (ICC), 2010 IEEE International Conference on*. 2010.

[23] Dong, N., N. Thinh, and Y. Xue, *Joint Network Coding and Scheduling for Media Streaming Over Multiuser Wireless Networks.* Vehicular Technology, IEEE Transactions on, 2011. **60**(3): p. 1086-1098.

[24] Mirkhosraie, S. and M. Hefeeda, *Live Peer-to-Peer Streaming with Scalable Video Coding and Network Coding*, in *In Proc. of ACM Multimedia Systems (MMSys'10)*. 2010: Scottsdale. p. 123-132.

[25] Kiraly, Z. and E.R. Kovacs, *Multi-Layered Video Broadcast using Network Coding and a Distributed Connectivity Algorithm*, in *Egerváry Research Group Seminar*. 2011, EGRES: H{1117, Budapest, Hungary.

[26] Chen, T., A. Bansal, and S. Zhong, *A reputation system for wireless mesh networks using network coding.* ELSEVIER, Journal of Network and Computer Applications, 2011. **34**: p. 545-541.

[27] Nguyen, K., T. Nguyen, and S.-C. Cheung, *Video Streaming with Network Coding.* Journal of Signal Processing Systems, 2010. **59**(3): p. 319-333.

[28] Cleju, N., N. Thomas, and P. Frossard, *Selection of network coding nodes for minimal playback delay in streaming overlays*, in *Multimedia (cs.MM); Information Theory (cs.IT); Networking and Internet Architecture (cs.NI).* 2011, IEEE Transactions on Multimedia: Cornell University.

[29] Wang, M. and L. Baochun. *Lava: A Reality Check of Network Coding in Peer-to-Peer Live Streaming*. in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. 2007.

[30] Wang, M. and B. Li, *R2: Random Push with Random Network Coding in Live Peer-to-Peer Streaming.* Selected Areas in Communications, IEEE Journal on, 2007. **25**(9): p. 1655-1666.

[31] Baochun, L. and N. Di, *Random Network Coding in Peer-to-Peer Networks: From Theory to Practice.* Proceedings of the IEEE, 2011. **99**(3): p. 513-523.

[32] Barekatain, B. and M.A. bin Maarof. *Network coding efficiency in live video streaming over Peer-to-Peer Mesh networks*. in *Information Technology in Asia (CITA 11), 2011 7th International Conference on*. 2011.

[33] Gajic, B., J. Riihijarvi, and P. Mahonen. *Performance Evaluation of Network Coding in Middle-Sized Networks*. in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. 2010.