Aalborg Universitet

PictureViewer

a Mobile Application Using Network Coding

Pedersen, Morten Videbæk; Heide, Janus; Fitzek, Frank; Larsen, Torben

PictureViewer
- A Mobile Application using Network Coding

Morten V. Pedersen
Dept. of Electronic Systems
Aalborg University
Email: mvpe@es.aau.dk

Janus Heide
Dept. of Electronic Systems
Aalborg University
Email: speje@es.aau.dk

Frank H.P. Fitzek
Dept. of Electronic Systems
Aalborg University
Email: ff@es.aau.dk

Torben Larsen
Dept. of Electronic Systems
Aalborg University
Email: tl@es.aau.dk

Abstract—**This paper looks into the implementation details of network coding for a mobile application running on commercial mobile phones. The mobile application coined PictureViewer can convey pictures from one source device to many neighboring devices using WiFi. The advantage of network coding in this context is that the source devices only needs a minimal amount of knowledge about the targets received packets and therefore only a minimal amount of feedback is needed to ensure reliable data delivery. The implemented network coding algorithms are tailored to be fast and energy efficient on commercial mobile phones. The goal of the paper is therefore to investigate those algorithms and to demonstrate that network coding can be deployed on state of the art mobile phones.**

Index Terms—**Energy, IEEE802.11b/g, mobile phones, network coding, performance, wireless networks.**

I. INTRODUCTION

Network coding has received a lot of attention since it was introduced in 2002 by Ahlswede et al. [1]. Several research works have investigated [2], [3] and implemented [4], [5] network coding to prove the feasibility of this novel technique. Network coding can be applied in many mobile communication scenarios such as multicast or meshed networking. Where network coding delivers promising results for throughput and reliability. While some codes are working end to end, in network coding packets can be recoded at each node in the network if needed. This can be of special interest in multi-hop networks.

Even though the concept of network coding has been proven to work, the current questions are how to design network coding algorithms and if these algorithms are too complex for a given platform. In [6], [7] we have shown that network coding can be applied to sensor networks and meshed networks formed by mobile phones. One finding was that network coding techniques must be designed with care if they are to be applied to the mobile or embedded domain. These platforms have limited resources such as energy, memory, and computational power in addition to the general problems in mobile networking such as limited wireless capacity.

This paper introduces a commercial mobile application using network coding that is running on the Symbian/S60 platform used on most Nokia smartphones and by other manufactures such as Motorola, Samsung, and Sony Ericsson. The main idea is that users wish to share content over short range technologies such as WiFi. Instead of uploading the content to social networks such as MySpace or Facebook, the content can be conveyed directly to mobile phones in the vicinity. This could be to show new photos among friends without being forced to look at a single small screen, instead everybody could use their own device.

The use of network coding is motivated by the fact that the transmission from one source to many sinks must be done in a reliable and efficient manner. Network coding enables this as it allows for efficient spectrum usage and a low complexity error control system. Additionally network coding offers an extension of the present idea in multi-hop networking where the content can be relayed to extend the coverage. Network coding can be applied at different protocol layers ranging from physical layer over network layer to application layer. In this work we will focus on the application layer. Furthermore the paper gives some implementation guidance on how to keep the complexity of network coding low. The performance evaluation metrics are the wireless capacity used, the time needed for exchange, and the energy consumed.

The remainder of this work is organized in the following sections. Section II introduces different transmission approaches. Section III describes the two network coding algorithms used in PictureViewer. In Section IV the functionality and interface of PictureViewer is introduced. Section V presents the results obtained with the application. The final conclusion is drawn in Section VI

II. TRANSMISSION APPROACHES

Different approaches for transmitting the data are possible, here we present some possibilities. We assume that a single source s broadcast data to N sinks $t_1...t_N$ and that the source has a direct wireless link to the sinks, as shown in Figure 1. The source has stored the pictures that must be conveyed to the sinks. Each picture is represented by a number of packets, n. Transmitting packets over the wireless link may lead to packet loss due to the characteristics of the wireless channel thus an error control system is needed.

A. Unicast

The simplest solution is for the source to send the picture in a round robin fashion using a reliable unicast protocol e.g. Transmission Control Protocol (TCP). Such an approach is fully reliable as each sink is served individually. Each sink
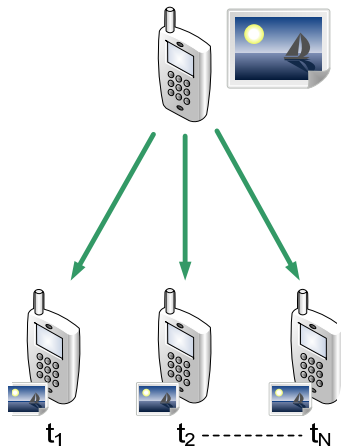
Fig. 1. The basic PictureViewer setup.

acknowledges each received packet and therefore the source device can determine when all sinks have received all packets. This solution is simple to implement and the computational complexity is low. However if the number of sinks is high it may take a long time to transmit a picture to all sinks. The energy consumption of the devices in this scenario is coupled with the time. The source will suffer as it has to transmit the picture N times. Additionally, the wireless capacity is affected as each additional sink will result in the use of additional spectrum.

B. Broadcast

Instead of sending to each device individually the source could broadcast the picture to the receiving nodes. This approach is highly efficient as long as no errors occur on the wireless link. However when packet losses occur some form of error correction is needed. To achieve reliability the source needs to know which packets have been lost by one or more sinks and thus needs to be retransmitted, this introduces the need for feedback information which consumes spectrum and time. The amount of feedback information depends on N and the packet loss probability. The feedback messages can be fairly small and as such they do not require a lot of spectrum, however they potentially introduces collisions in the network as both the source and sinks will attempt to transmit packets. Thus the performance of such a broadcast approach also depends on the effectiveness of the Medium Access Control (MAC).

Furthermore the retransmissions by themselves is suboptimal as not all sinks will lose the same packets, thus each retransmitted packet will only be useful for a subset of the sinks. E.g. if mobile devices 1, 2, and 3 have lost packet 17, 21, and 16 respectively, three broadcast packets must be transmitted. Each retransmitted packet is only useful for a single source, while the initial packet transmission is useful for all sinks. Generally broadcast can be faster than unicast if $N > 1$ and its performance is less sensitive towards the number of sinks.

C. Pure Network Coding

One network coding approach that lends itself to this scenario, is Random Linear Network Coding (RLNC) [2]. With this approach coding is used to simplify the problem of correcting lost packets at the sinks and furthermore reduces the requirement for feedback from the sinks. In network coding nodes can combine the information in the network to create new packets [8]. Hence the source codes $n + r$ packets from the n original packets and broadcasts these packets. r is the number of redundant packets and should be chosen according to the Packet Error Probability (PEP) of the link. Each sink only has to receive any n linear independent packets, which can then be decoded to recreate the original packets.

The advantage of network coding can be illustrated by the previous example. In this case the source could code packets 16, 17, and 21 together into a new packet of the same length as the original packets. This packet is broadcasted to the three sinks, which each remove from the coded packet the packets they already got and thus decode the packet into the packet they lost. Thus the retransmission that needed three transmissions using broadcast can be done by a single transmission using network coding.

As the coding and decoding operations introduces complexity the computational requirement is increased. These operations will increase the Central Processing Unit (CPU) load and thus the energy consumption. However the number of redundant packets transmitted from the source and feedback messages sent from the sinks can be decreased which will help to decrease energy consumption.

D. Systematic Network Coding

To decrease the complexity systematic network coding can be used [9]. Systematic network coding combines the broadcast and network coding approaches. As there is no obvious gain in coding the first n packets, the source broadcasts these packets and code the remaining r packets. Each uncoded packet is useful for all N sinks as they are linear independent. The following r packets are coded and have a high probability of being independent of the n uncoded packets. This approach decreases the computational complexity at the source and the sinks as only r packets has to be coded and decoded.

	Complexity	Delay	Energy	Capacity
Unicast	Low	High	High	Low
Broadcast	Low	Medium	Medium	Medium
Pure coding	High	Medium	Medium	High
Systematic coding	Medium	Low	Medium	High

TABLE I
ESTIMATES OF THE COMPUTATIONAL COMPLEXITY, DELAY, ENERGY CONSUMPTION AND ACHIEVABLE CAPACITY WHEN $N >> 1$.

The different approaches are compared in Table I. In the following we implement and compare network coding and systematic network coding.

III. Coding Algorithms

Here we briefly introduce the two implemented network coding algorithms used in PictureViewer, for details and analysis see [10]. We base our solution on performing RLNC over a Galois field. When Galois fields are implemented on computer systems the Galois elements are generally of the form 2^i, where $i \in \mathbb{Z}^*$, and typically $i \in \{8, 16, 32\}$. We choose the smallest possible Galois Field, GF(2), to decrease the computational complexity of coding operations. This is done to overcome the challenges posed by the limited computational resources available on the test platform.

The data to be transferred from the source, to the sinks is divided into packets of length m. The number, n, of original packets over which encoding is performed is typically refereed to as the batch size or generation size. Thus the n original data packets of length m are arranged in the matrix $\mathbf{M} = [\mathbf{m}_1\mathbf{m}_2 \ldots \mathbf{m}_n]$, where $\mathbf{m}_i$ is a column vector.

In pure network coding to generate one coded data packet $\mathbf{x}$, $\mathbf{M}$ is multiplied with a randomly generated vector $\mathbf{g}$ of length n, $\mathbf{x} = \mathbf{M} \times \mathbf{g}$. In this way we can construct $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_{n+r}]$ that consists of $n + r$ coded data packets and $\mathbf{G} = [\mathbf{g}_1\mathbf{g}_2 \ldots \mathbf{g}_{n+r}]$ that contains $n + r$ randomly generated encoding vectors. In order for a sink to successfully recreate the original data packets, it must receive n linear independent coded packets and encoding vectors. All received coded packets, $\hat{\mathbf{x}}_i$, are placed in the matrix $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1\hat{\mathbf{x}}_2 \ldots \hat{\mathbf{x}}_n]$ and all encoding vectors, $\hat{\mathbf{g}}_i$, are placed in the matrix $\hat{\mathbf{G}} = [\hat{\mathbf{g}}_1\hat{\mathbf{g}}_2 \ldots \hat{\mathbf{g}}_n]$. The original data $\mathbf{M}$ can then be decoded as $\hat{\mathbf{M}} = \hat{\mathbf{X}} \times \hat{\mathbf{G}}^{-1}$.

In systematic network coding both uncoded and coded packets are generated. Uncoded packets can also be perceived as coded packets with a trivial encoding vector where a single element in the encoding vector $\mathbf{g}$ is one and all other, $n - 1$, elements are zero. Thus we can generate an uncoded packet $\mathbf{y}$ from its trivial encoding vector $\mathbf{h}$, $\mathbf{y} = \mathbf{M} \times \mathbf{h}$. In this way we can construct $\mathbf{Y} = [\mathbf{y}_1\mathbf{y}_2 \ldots \mathbf{y}_n]$ that consists of n uncoded data packets and $\mathbf{H} = [\mathbf{h}_1\mathbf{h}_2 \ldots \mathbf{h}_n]$ that contains the n independent trivial encoding vectors. Furthermore we construct $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_r]$ that consists of r coded data packet and $\mathbf{G} = [\mathbf{g}_1\mathbf{g}_2 \ldots \mathbf{g}_r]$ that contains r randomly generated encoding vectors. For a sink to successfully recreate the original data packets, it must receive n linear independent coded packets and encoding vectors. n received uncoded $\hat{\mathbf{y}}_i$ and coded $\hat{\mathbf{x}}_i$ packets are placed in the matrix $[\hat{\mathbf{Y}}\hat{\mathbf{X}}] = [\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2 \ldots \hat{\mathbf{y}}_{(n-i)} \ \hat{\mathbf{x}}_1\hat{\mathbf{x}}_2 \ldots \hat{\mathbf{x}}_i]$ and the n corresponding encoding vectors are placed in the matrix $[\hat{\mathbf{H}}\hat{\mathbf{G}}] = [\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2 \ldots \hat{\mathbf{h}}_{(n-i)} \ \hat{\mathbf{g}}_1\hat{\mathbf{g}}_2 \ldots \hat{\mathbf{g}}_i]$. The original data $\mathbf{M}$ can then be decoded as $\hat{\mathbf{M}} = [\hat{\mathbf{Y}}\hat{\mathbf{X}}] \times [\hat{\mathbf{H}}\hat{\mathbf{G}}]^{-1}$.

Encoding using GF(2) is implemented by bitwise xoring the packets together for which the corresponding indices of the encoding vector is one. Decoding can be performed in several ways, we have chosen Gauss-Jordan elimination because the data is then always maximally decoded. This ensures that the load of decoding is done on the fly and not at the end when enough packets have been received.

To determine the synthetic performance of the two low complexity network encoding algorithms we have implemented

and tested them on a commercially available mobile phone, see Figure 2. The Nokia N95-8GB is a fast cutting edge mobile phone with the following specs; ARM 11 332 MHz CPU, 128 MB RAM, Symbian OS 9.2. In this test a single phone encode packets save them to memory and subsequently decodes them, thus the network is not utilized. For the pure network coding n coded packets will be generated and subsequently decoded. For the systematic network coding the first $0.7 \cdot n$ of the packets are uncoded and the last $0.3 \cdot n$ packets are coded. Thus the coding performance corresponds to what would be expected if the packets were transmitted over a channel with PEP=0.3.
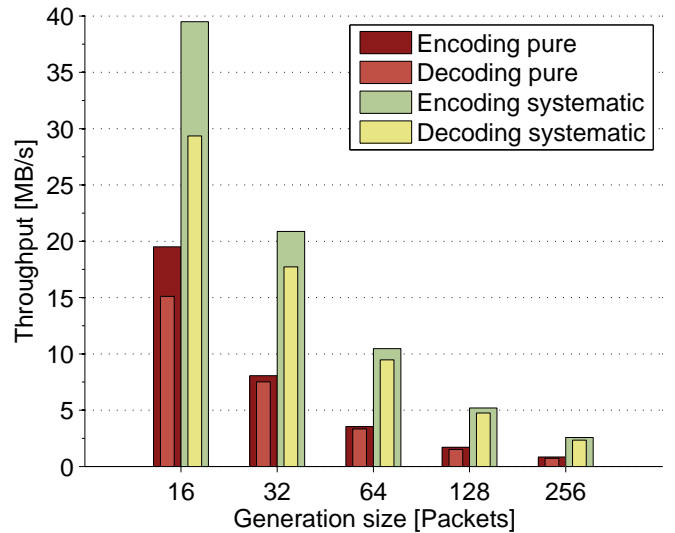

Fig. 2. Synthetic throughput for encoding and decoding.

As seen in Figure 2 the encoding and decoding speed decreases as the generation size increases. Additionally the decoding throughput is somewhat lower than the encoding throughput. When working in GF(2) the number of operations needed for encoding and decoding is the same. Thus it should be possible to achieve similar encoding and decoding throughput. However, the implementation of the decoding algorithm is slightly less straightforward than the encoding algorithm, and is therefore suboptimal in the current implementation. In this test the systematic approach achieves twice as high throughput compare to the pure approach for a generation size of 16. For generation size of 64 and above the throughput is tripled. For a small generation size memory allocations and other constant contributions is important, while at large generation size the CPU is the dominant bottleneck. The performance also depends on the ratio between uncoded and coded packets and in the worst case where all packets are coded the two approaches perform identically and thus have the same throughput. Thus the systematic approach is useful if a high throughput and low energy consumption are important parameters.

IV. Implementation

This section introduces the PictureViewer application. The PictureViewer application allows users to broadcast images
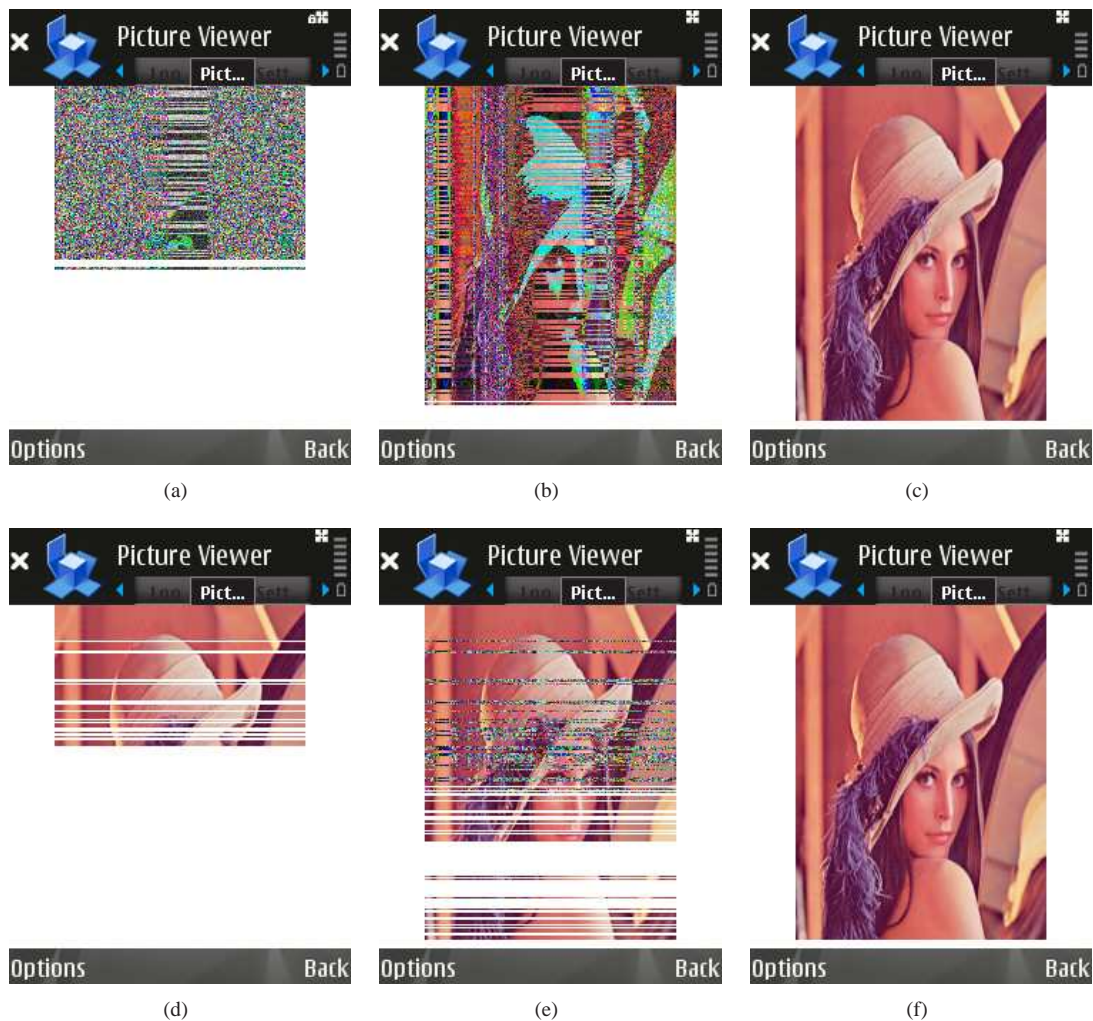
Fig. 3. Pure network coding; (a) Partially decoded data, (b) Image starting to appear as the decoders rank increases, (c) The final decoded image. Systematic network coding; (d) Received uncoded data, (e) Erasures corrected by coded packets, (f) The final decoded image.

located on their phones to a number of receiving devices. The application uses the coding algorithms described in the previous section, and can be configured to operate using either the pure network coding strategy or the systematic strategy. To illustrate the difference between these strategies the PictureViewer application allows users to monitor the decoding process directly. The decoding process is displayed by drawing the actual content of the decoding matrix onto the display of the receiving phones. In Figure 3 the first row of screenshots shows the decoding process of the pure network coding approach. In this approach only coded packets are transmitted and all packets contained in the decoder matrix are linear combinations of other packets. Initially as shown in Figure 3(a) the content of the decoding matrix looks like noise. However, as the decoder receives more linear combinations and the decoding process starts to solve the decoding matrix, the original picture start to appear as the pixel values approach the solution, see Figure 3(b). In the final Figure 3(c) the picture has been decoded and the transmission is complete. In Figure 3 the second row of screenshots shows the systematic strategy.

Using this strategy we first convey the full data set uncoded. Figure 3(d) shows how uncoded packets are being inserted into the decoding matrix, this continues until all original packets have been sent once. In Figure 3(e) we have entered the coding phase, in this phase erasures which occurred during the uncoded phase are being repaired by transmitting encoded packets. In this test the PEP was approximately 30% and therefore we received 70% of the data uncoded without the need for any additional decoding computations. This illustrates the advantage of the systematic approach as the number of packets that had to be decoded was reduced by 70%. In the current implementation PictureViewer ensures a high probability of decode-ability by always sending a fixed overhead of encoded packets, this is obviously not the optimal strategy and future work should investigate the development of an efficient feedback protocol.

V. RESULTS

In this section we present results obtained with the Picture-Viewer application running on two Nokia N95-8GB mobile

phones. One phone acts as the source while the other phone is the sink. In the current implementation no feedback is transmitted from the sink to the source and thus an arbitrary number of additional sinks can be added without any degradation of performance.

To increase the accuracy of the measurements the source transmit a 5 MB file instead of a picture to increase the time span of each measurement. Performance have been measured at different generation sizes to determine how the increase in complexity when the field size increases, influences the performance of the application.

The performance when data is not coded has been added for reference, as it is not directly comparable to the performance when coding is performed. The reason is that the current implementation does not incorporate feedback and thus traditional broadcast is not practical as each packet would have to be transmitted many times to ensure that it reaches the sink. However the results without coding are interesting as they indicate, the top speed at which the phone can broadcast, how much broadcast by itself will load the CPU, and the power consumption during broadcast.
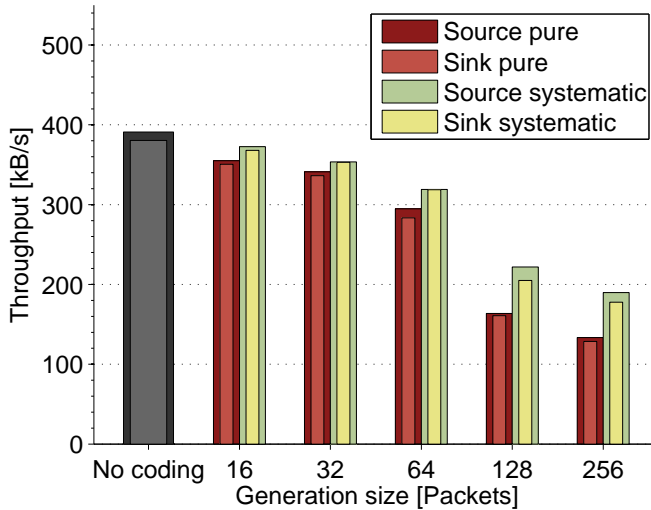

Fig. 4. Throughput.

In Figure 4 we see that the throughput decreases as the generation increases. This is not surprising if we recall the results from Figure 2 which show the raw encoding and decoding throughput as a function of the generation size. For low generation size the throughput of coding is similar to that of no coding this indicates that the computational overhead in this case is not the bottleneck. As the generation size increases the coding operations becomes more computational demanding and the throughput decreases. Additionally we observe that the systematic approach achieves higher throughput especially for higher generation sizes. This is expected as the systematic approach increases the coding throughput which is the limiting factor at high generation sizes.

Figure 5 shows the CPU load of the source and sink during the test. It is interesting to observe that transmitting uncoded
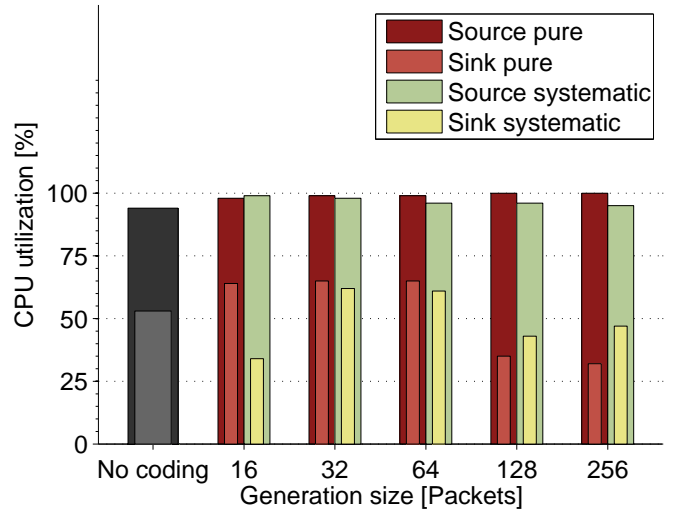

Fig. 5. CPU utilization.

data results in a CPU load of almost 100%. This indicates that the transmission speed is limited by the CPU rather than the wireless interface. Thus the added load from coding operations is expected to decrease the transmission throughput, which we observed in Figure 4. Note that the CPU load at the sink is higher for the systematic approach when the generation size is high. The reason is that the systematic approach allows the source to transmit at a higher rate and thus the sink processes more packets per time unit.
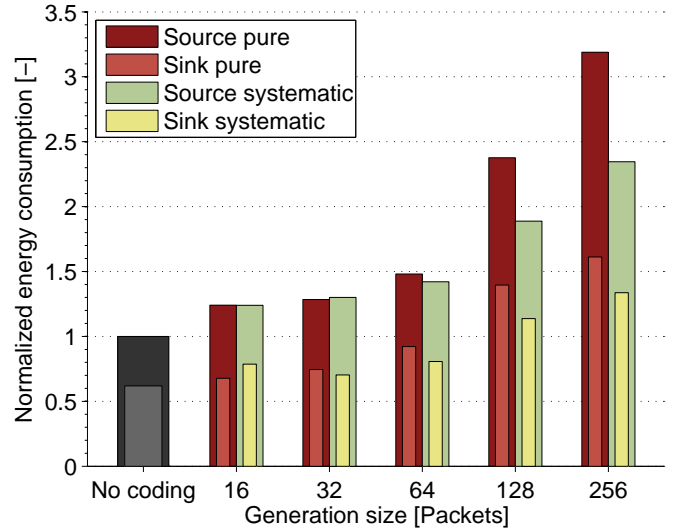

Fig. 6. Normailzed energy consumption.

In Figure 6 we observe that the energy consumption increase with the generation size. The reason is that a higher generation size results in lower throughput and resulting longer transmission time. The systematic approach decreases this effect by reducing the complexity and thus decreasing the energy consumption.

VI. Conclusion

In this paper we have introduced a mobile application called PictureViewer that via network coding enables a user to share content with several other users. The application itself is simple but it demonstrates that network coding does not necessarily result in high complexity or overwhelming energy consumption. The implemented algorithms are designed to allow for high coding throughput. This was achieved by using a binary Galois Field and a systematic random code. The measurements presented in this work confirm that the algorithms are fast and that the systematic approach outperforms the pure network coding approach.

PictureViewer visualizes encoding and decoding through pictures and thus is a tool for demonstrating network coding. Additionally PictureViewer can serve as a platform for investigating feedback approaches that allow for fully reliable broadcast. Another possibility is to deploy PictureViewer in a multi-hop mesh network where recoding and forwarding approaches could be investigated.

References

[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] T. Ho, R. Koetter, M. Medard, D. Karger, and M. ros, "The benefits of coding over routing in a randomized setting," in *Proceedings of the IEEE International Symposium on Information Theory, ISIT '03*, June 29 - July 4 2003.

[3] M. Médard and R. Koetter, "Beyond routing: An algebraic approach to network coding," in *INFOCOM*, 2002.

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '06)*, pp. 243–254, ACM Press, September, 11-15 2006.

[5] J.-S. Park, M. Gerla, D. S. Lun, Y. Yi, and M. Medard, "Codecast: a network-coding-based ad hoc multicast protocol," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 13, no. 5, pp. 76–81, October 2006.

[6] R. Jacobsen, K. Jakobsen, P. Ingtoft, T. Madsen, and F. Fitzek, "Practical Evaluation of Partial Network Coding in Wireless Sensor Networks," in *4th International Mobile Multimedia Communications Conference (MobiMedia 2008)*, (Oulu, Finland), ICTS/ACM, July 2008.

[7] J. Heide, M. V. Pedersen, F. H. Fitzek, and T. Larsen, "Cautious view on network coding - from theory to practice," *Journal of Communications and Networks (JCN)*, 2008.

[8] C. Fragouli, J. Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63–68, 2006.

[9] M. Xiao, T. Aulin, and M. Médard, "Systematic binary deterministic rateless codes," in *Proceedings IEEE International Symposium on Information Theory*, July 2008.

[10] J. Heide, M. V. Pedersen, F. H. Fitzek, and T. Larsen, "Network coding for mobile devices - systematic binary random rateless codes," in *The IEEE International Conference on Communications (ICC)*, (Dresden, Germany), 14-18 June 2009.