Cautious View on Network Coding - From Theory to Practice

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

Published in:
Journal of Communications and Networks

Publication date:
2008

Document Version
Early version, also known as pre-print

Cautious View on Network Coding – From Theory to Practice

Janus Heide, Morten V. Pedersen, Frank H.P. Fitzek, and Torben Larsen

Abstract: **Energy consumption has been mostly neglected in network coding (NC) research so far. This work investigates several different properties of NC that influence the energy consumption and thus are important when designing NC systems for battery-driven devices. Different approaches to the necessary implementation of coding operations and Galois fields arithmetic are considered and complexity expressions for coding operations are provided. We also benchmark our own mobile phone implementation on a Nokia N95 under different settings. Several NC strategies are described and compared, furthermore expressions for transmission times are developed. It is also shown that the use of NC introduces a trade off between reduction in transmission time and increase in energy consumption.**

Index Terms: **Energy, IEEE802, mobile phones, network coding, performance, wireless networks.**

I. INTRODUCTION

In this work, we focus on problems and challenges posed by implementing random linear network coding (RLNC) on commercially available mobile devices and how this affects the utilization of network interfaces. RLNC is considered as it is a promising candidate for a practical implementation of network coding (NC) and has properties that allow for low complexity control and error correction systems [1]. We consider RLNC for deployment in a wireless network and it is therefore natural that the implementation is tested on mobile devices. Such devices are most often battery driven and the energy consumption therefore influences the operational time, which is an important factor for the end-user. Thus an efficient utilization of wireless interfaces becomes an important design criterion as it can constitute a considerable portion of the energy consumption. Another important factor is the additional computational resources spent as a result of coding operations, this will inevitable increase the energy consumption.

A large body of existing literature [2] treats the benefits of NC that can be shown theoretically. The costs of implementing NC in terms computational overhead, memory consumption or network usage is often not considered. However, the attention towards practical implementation problems of NC is on the rise and is increasingly relevant as deployment schemes are being proposed and evaluated [3]–[6]. In [7] and [8], implementation problems are investigated with focus on the computational complexity of coding operations. An implementation using log and anti-log tables is constructed and evaluated. Throughput up to 11 MB/s is reported on a 3.6 GHz dual core Intel P4 central processing unit (CPU), but performance decreases fast as the number of packets that are coded together increases. Different optimization techniques are described and incorporated. Generated network overhead as a consequence of utilizing NC is commented and sought to be reduced. Furthermore, the concepts of *aggressiveness* and *density* and its impact of performance are studied. In [9], a coding throughput of 44 Mb/s is reported using an implementation based on a simple full-size look-up table. This is achieved on a 800 MHz Intel Celeron CPU when 32 packets, $K = 32$, of 1500 B are coded together. The observed throughput is approximately 10 times higher than that reported in [7] at similar settings. The authors conclude that deployment is not a problem, however it is mentioned that throughput for $K = 32$ is the best case, unfortunately no performance for $K > 32$ is provided. In [10], tables are not used for Galois field multiplication, instead a loop based approach is used. In combination with single instruction, multiple data (SIMD) instruction sets this approach provides a large performance increase of over 500% compared to a baseline implementation. The implementation is further optimized through parallelization and encoding speeds up to 43 MB/s is reported on a 2.8 GHz quad core Intel P4 CPU. Even though these results are promising, they do not necessarily indicate that NC can be deployed on mobile devices, such as mobile phones or wireless sensors, because the hardware resources available on such devices are considerably lower than those of the desktop computers used to obtain these results. However the throughput needed on such mobile devices can also be assumed to be considerably lower than that of a desktop computer. One additional constraint that these evaluations do not consider is the energy consumption which is an important factor for mobile devices. In spite of the large amount of research related to NC very little attention has been directed towards NCs effect on the energy consumption. In [11] and [12], the problem of minimum energy broadcasting in ad-hoc wireless networks is treated. Although this work is obviously related to energy consumption, it focuses on the network layer and the layers below and takes a theoretical approach by relying on analysis and simulation. Our work is directed towards the application layer and considers implementation and real life evaluation. To the best of our knowledge there is not any existing work that is directly related to the work presented here.

Our contribution is an overview and discussion of practical implementation of NC and is two-fold. First we consider the *internal* aspects of NC on a mobile device, which is related to what is executed on the device and its hardware limitations. Secondly we provide an *external* view, which is related to the properties of the communication links and the network topology. As we primarily consider deployment on mobile devices our main focus is on energy consumption and how it is affected when NC is

Manuscript received July 21, 2008.

The authors are with the Department of Electronic Systems, Aalborg University, Denmark, email: {speje, mvpe, ff, tl}@es.aau.dk.

1229-2370/08/$10.00 © 2008 KICS

used in a wireless network. As coding operations consume computational resources and in turn those increase the energy consumption, we consider the computational complexity of RLNC and discuss different implementation approaches. We provide benchmarks of encoding and decoding on our own implementation running on a high-end mobile phone, the Nokia N95. We describe different NC strategies and their operation in detail for two example networks and give expressions for the transmission time of the different strategies. We also show that there exists a trade off between energy consumption and transmission time when NC is deployed in such networks. These findings show that additional considerations exist when NC is employed on battery driven devices and that new strategies therefore must be developed and evaluated both with respect to throughput and energy consumption.

This work is organized in the following sections. Section II-A presents an analysis of the computational complexity and its implications. In Section III-A, the transmission time and energy consumption of different transmission strategies is presented in a simple scenario. In Section III-B, the example is expanded to a scenario that is better suited for demonstrating wireless transmission and the strategies is described in more detail. The final conclusion is drawn in Section IV.

II. INTERNAL VIEW

In current research network coding is being evaluated to enhance a wide variety of protocols, services and applications. To utilize network coding, nodes in the network will have to perform coding operations on the data-flows traversing them. These operations incur additional resource consumption on the devices and can potentially define a lower bound on the type of devices capable of utilizing these new ideas. In RLNC, one concern is the high computational complexity [7], which limits the currently achievable coding bandwidth. This poses not only a problem for low-performance devices, but also battery driven mobile devices where high CPU utilization affect the standby and operational time negatively. In the following, we will determine the computational complexity of performing RLNC and investigate the cost of the underlying algebraic finite field operations. These operations can be implemented in a number of ways, with various advantages and drawbacks, which will be presented. Finally a number of measurement results will be presented showing the maximum coding bandwidth and energy consumption of a specific RLNC implementation on high-end mobile devices.

A. Computational Complexity

In RLNC, source nodes are encoding data packets to be transmitted onto the network. This operation must be reversed at the sink nodes, by decoding the received packets. These operations require a certain amount of computational effort. To determine if the deployment of RLNC is feasible on devices with low computational resources we consider the computational complexity of these operations. For a more detailed description of the encoding and decoding operations, consult [13].

A.1 Encoding

In RLNC, two factors influence the computational complexity of the encoding process, the first is the number of original packets, n, used to generate one encoded packet. In RLNC, this number is commonly refereed to as the size of the batch or generation. The second factor is the length, m, of the individual packets, in this context the length is not necessarily counted in bytes, but dependent on the element size of the underlying algebraic finite field. Commonly finite field elements represents 8, 16, or 32 bits, e.g., if each field element is 16 bits and a packet is 1200 bytes the resulting length is 600 elements.

To generate one encoded data packet $\mathbf{x}$, n original data packets of length m are arranged in the matrix $\mathbf{M} = [\mathbf{m}^1\mathbf{m}^2 \cdots \mathbf{m}^n]$ and multiplied with a randomly generated vector $\mathbf{g}$ of length n. The encoding can thus be written as a simple matrix multiplication, $\mathbf{X} = \mathbf{M} \times \mathbf{G}$, where the result $\mathbf{X} = [\mathbf{x}^1\mathbf{x}^2 \cdots \mathbf{x}^n]$ consists of the n encoded data packets and $\mathbf{G} = [\mathbf{g}^1\mathbf{g}^2 \cdots \mathbf{g}^n]$ contains n randomly generated encoding vectors. The complexity of generating a random number is system specific but is assumed to be constant, r, hence generating $\mathbf{g}$ has the complexity $\mathrm{O}(nr)$. The matrix-vector multiplication has the complexity $\mathrm{O}(nm)$, when naive multiplication is used [14] and encoding one packet $\mathbf{x}$ therefore requires $\mathrm{O}(n(r+m))$. In order for a receiver to decode n original packets it must receive n linear independent encoded data packets. Encoding n packets from $\mathbf{M}$ has the complexity $\mathrm{O}(n^2(r+m))$.

A.2 Decoding

In order for the decoder to successfully recreate the original data packets, n linear independent encoded packets and encoding vectors must be received. All received encoded packets are placed in the matrix $\mathbf{X} = [\mathbf{x}^1\mathbf{x}^2 \cdots \mathbf{x}^n]$ and that all encoding vectors are placed in the matrix $\mathbf{G} = [\mathbf{g}^1\mathbf{g}^2 \cdots \mathbf{g}^n]$. The original data $\mathbf{M}$ can then be decoded as $\mathbf{M} = \mathbf{X} \times \mathbf{G}^{-1}$. Decoding one packet is performed by the matrix-vector multiplication $\mathbf{X} \times (\mathbf{g})^{-1}$, with complexity $\mathrm{O}(nm)$, however this requires that the encoding vector $\mathbf{g}$ has been inverted. Inverting $\mathbf{G}$ has the complexity $\mathrm{O}(n^3)$, when Gauss-Jordan elimination is applied. Thus, decoding n encoded packets has the complexity $\mathrm{O}(n^2m + n^3)$. It should be noted that the computational complexity can be lowered slightly as some lower complexity matrix-matrix multiplication and matrix inversion methods exist [14].

As seen the encoding and decoding operations consist mainly of matrix multiplications and inversions of dense matrices with makes the used algorithms difficult to optimize. Even though some optimizations exist the resulting improvements are minor, and usually apply only to very large matrices [15]. Since the amount of operations needed cannot be reduced significantly, further optimizations must target the individual matrix element operations.

B. Finite Fields Arithmetics

The algebraic operations used in RLNC encoding and decoding are defined within a branch of mathematics known as a finite fields or Galois fields. Galois fields are special constructs consisting of a finite number of elements, over which standard

Table 1. Memory consumption of log and anti-log tables used for Galois field arithmetic's using different field sizes.

Table type	Field size	Memory consumption
Logarithmic	GF(2^8)	512 B
Logarithmic	GF(2^{16})	128 KB
Logarithmic	GF(2^{32})	4.2 GB

arithmetic operations (addition, subtraction, multiplication, and division) are closed. This essentially means that any arithmetic operations performed on two elements in the field will result in an element also in the field. As implementations of various coding algorithms have to deal with the finite precision of computers, Galois fields have become a key component when implementing arithmetic operations required by e.g., forward error correction (FEC) or cryptographic algorithms. As all arithmetics with packets performed in RLNC are computed over a Galois field, the performance of the Galois field implementation is a critical factor in the overall system performance. Especially multiplication and division are computationally demanding and therefore often optimized using logarithmic look-up tables. A comparative study of different techniques is presented in [16], in which Greenan *et al.* find that the performance of different Galois field implementations is highly dependent on the underlying hardware. Different platforms might therefore benefit from different representations of the lookup tables. In a recent paper [10] by Shojania *et al.*, a standard logarithmic table based implementation is out-performed by a loop based approach, implemented using hardware accelerated instructions available in most modern CPUs. This approach also alleviates the memory consumption imposed by the table base approaches, which could pose a problem on memory constrained devices. To indicate the memory overhead using tables, the memory consumption of the widely used logarithmic table based method is presented in Table 1.

As seen, tables representing fields larger than 16 bit are hardly acceptable even on high-end devices. However implementations using 8 or 16 bit elements will not take advantage of the full 32 or 64 bit data-path on most modern hardware [17]. This influences performance since the bits constituting a packet is represented by a number of Galois field elements. Using large fields have the additional advantage of reducing the number of elements and thereby calculations significantly. As an example moving from an 8 bit field to a 32 bit field reduces the number of calculations by 75%.

C. RLNC Implementation

In the following, we present the results from the first implementation of RLNC on a high-end mobile phone, namely the Nokia N95.

The Nokia N95 is one of the fastest mobiles currently available equipped with an ARM 11 CPU running at 332 MHz [18]. These results will serve as an initial indication of the performance of a basic RLNC implementation, using standard matrix calculations and Galois field implemented by logarithmic table lookup. Fig. 2 shows the encoding and decoding throughput measured for two different field sizes, in all tests a total of 200 KB was coded.

Fig. 1. The testbed of Nokia N95 mobile phones.

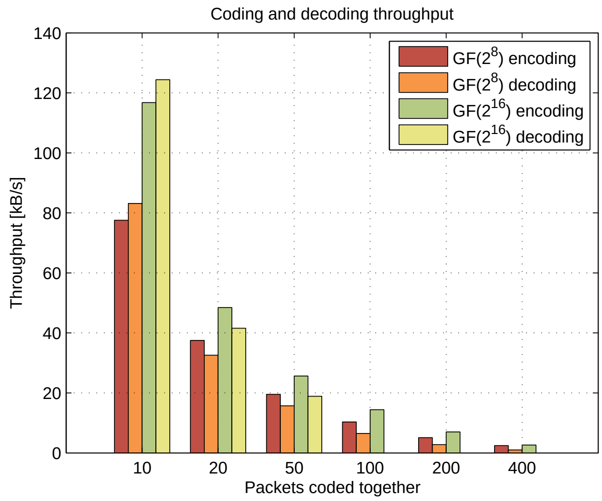
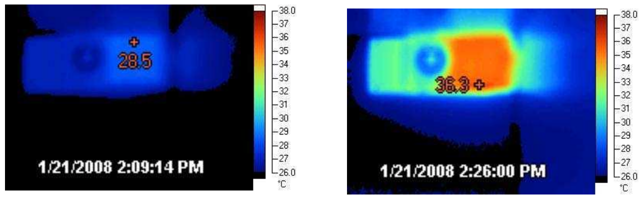


Fig. 2. Measured throughput of our RLNC implementation on a Nokia N95 using underlying GF(2^8) and GF(2^{16}).

Using GF(2^8) the highest measured encoding throughput was 78 KB/s which is slightly slower than decoding that reached 83 KB/s. The coding throughput drops quickly as the number of packets coded increases. The general slower decoding is a result of the higher computational complexity of decoding, due to the additional matrix inversion. Using GF(2^{16}) results in increased throughput with the highest measured encoding throughput reaching 117 KB/s while decoding reached 124 KB/s.

Clearly from these results, this type of basic implementation yields far from impressive throughput and optimizations is needed to match the wireless local area network (WLAN) speeds achievable on the N95. The low performance of RLNC has an additional cost on this type of devices, namely a substantially increased energy consumption, which during encoding and decoding was measured to 0.6 W. As a comparison this is approximately half the energy spent while using WLAN to receive data. This kind of energy consumption would result in a noticeable degradation of the phones standby and operational time. More efficient RLNC implementations is therefore needed. Software optimization has shown to improve the per-



(a) (b)

Fig. 3. Thermal images of the Nokia N95: (a) WLAN is powered down and (b) WLAN has been powered on for approx. 15 minutes and the device is considerably heated from power dissipated from the WLAN card.

formance considerably which would also decrease the energy consumption. Another possibility is hardware accelerated approaches, such solutions are still to be seen but could potentially reduce energy consumption further.

III. EXTERNAL VIEW

In this section, we consider how the utilization of network coding affects the energy consumption of a mobile device. Energy consumption is becoming an increasingly important topic for mobile device manufactures as they face several conflicting requirements from the users. On one side users demands new power hungry features such as global positioning system (GPS), large screens and fast wireless network connections while on the other side they want small devices with long standby and operational time. However as the capacity of batteries is not subject to the same exponential growth as processing power and storage [19], energy awareness is becoming an important parameter in the design of new features. Consequential researchers may have to move focus from performance in terms of throughput towards performance in terms of energy consumption, when future wireless systems are designed. In [20], it is estimated that communication and signal processing capabilities constitute roughly 50% of the total power budget for a device. Therefore a reduction in the power consumption of these capabilities can have a substantial impact on the overall power usage. Simply using higher capacity batteries will not necessarily eliminate this problem as high power consumption can become problematic as it may result in overheating of devices [21]. Fig. 3 shows thermal images of a Nokia N95 which illustrates the impact of heating due to the power dispatched in the internal WLAN card.

In [22], it was shown how NC can decrease the number of transmissions necessary to distribute data between a number of devices and thereby increases the throughput of the individual links. To calculate the "coding gain" for different topologies, a number of simplified communication scenarios was proposed and analyzed. In the following we base our studies on one of these topologies, namely the *wheel*, in order to investigate the "energy gain."

In the *wheel* example network, pairs of nodes attempt to exchange data via a single center node, thus coding opportunities exist in this network. In order to calculate and compare the energy consumption of different communication approaches, we have measured the power consumed on a mobile device, in this case the Nokia N95, during different operation modes using WLAN [23].

Table 2. Power levels used in broadcast - 1000 byte.

State	Power [W]	Data rate [Mbps]	Loss rate [%]
Sending	$1,629$	$5,623$	–
Receiving @ 3m	$1,375$	$5,379$	$3,328$
Receiving @ 30m	$1,213$	$5,115$	$8,324$
Idle @ 3m	$0,979$	–	–
Idle @ 30m	$0,952$	–	–
Sleep	0	–	–

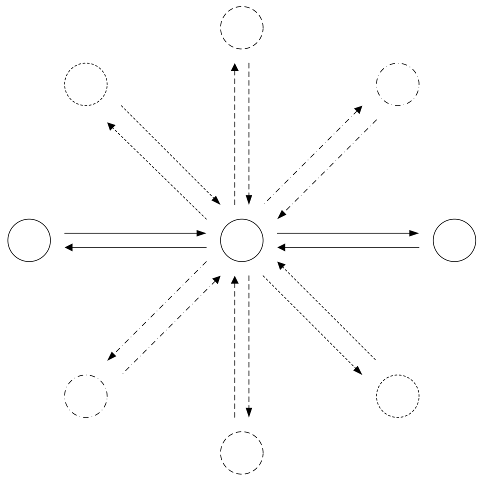


Fig. 4. The *wheel* example network.

To offer an alternative view we also consider technologies where the idle power consumption is reduced to near zero e.g., digital video broadcasting-handheld (DVB-H). This will change the resulting energy consumption and is interesting because such technologies could become common in the near future.

A. The Wheel Network

We consider the wheel example as introduced in [22]. We use this well established example network to achieve a better understanding of how network coding influences the utilization of the nodes network interfaces. As the example network does not use a realistic channel model, the results should not be generalized to general wireless networks. Nevertheless the results are interesting as they demonstrate that deployment of network coding may be problematic, when the energy consumed by the wireless interfaces in such a network is considered.

In the wheel N, nodes are placed on a circle around a center node. The nodes on the circle will also be referred to as the outer nodes. Two on the circle diametrically oppositely located nodes is a pair and both nodes in such a pair wish to transmit one packet to its peer, see Fig. 4. The center node can receive and transmit packets to all nodes on the circle. Each of the outer nodes is within transmission range of all other nodes, except its peer.

All packets have the same size and are transmitted in slots. A slot is a fixed time in which one node can transmit one packet over the channel. We assume an ideal medium access strategy

Table 3. Pure relaying.

	Sending	Receiving	Idle
One outer node	1	1	$2(N-1)$
All outer nodes	N	N	$2N(N-1)$
Center node	N	N	0
Sum	$2N$	$2N$	$2N(N-1)$

without collision. For each slot each node can be in one of three states, sending, receiving, or idle. We consider three different transmission schemes, pure relaying, pure network coding, and network coding in teams.

A.1 Pure Relaying

To ensure that all node pairs exchange their packet, all outer nodes transmit their packet to the center node that forwards it to the destination. Hence, each outer node transmits once and the center node repeats those transmissions. Thus, the number of sending slots equals $2N$. As each transmitted packets is received by only one node, the number of receiving nodes is equal to the number of sending nodes. When one of the N outer nodes transmits, the remaining $(N-1)$ outer nodes are idle and the center node is receiving. When the center node is sending, one outer node is receiving and $(N-1)$ outer nodes are idle. Thus, the total number of idle nodes is given by $2N(N-1)$. The transmission time S is the time until the exchange of packets have been completed and is given by the number of sending slots, as only one node can transmit in each slot

$$S = 2N. \tag{1}$$

A.2 Pure Network Coding

Each outer node transmits its packet to the center node, this transmission is overheard by all outer nodes except the peer of the sending node. Hence, when all outer nodes have transmitted, the center node holds all N packets and all outer nodes holds $N-1$ packets. The center node then codes all N packets together and transmit the coded packet which is received by all outer nodes. This enables each outer node to decode the one packet it did not receive, because it now holds $N-1$ packets and a combination of all N packets. In this way the center node is never idle and the outer nodes are only idle when their peer is transmitting. The transmission time is given by the number of sending slots

$$S = N + 1. \tag{2}$$

Pure network coding decreases the transmission time, however the number of idle slots is reduced and the number of receiving slots is increased. If the power consumption while receiving is larger than the consumption while idle, this may result in higher energy consumption. We therefore propose an approach that decreases the number of receiving slots while still using network coding.

A.3 Network Coding in Teams

The pairs of nodes is divided into a number of teams T, hence a team constitute a subset of $N/2T$ pairs or N/T nodes. Only

Table 4. Pure network coding.

	Sending	Receiving	Idle
One outer node	1	$N-1$	1
All outer nodes	N	$N(N-1)$	N
Center node	1	N	0
Sum	$N+1$	N^2	N

Table 5. Network coding in teams.

	Sending	Receiving	Idle
One outer node	1	$\frac{N}{T}-1$	$N\frac{T-1}{T}+T$
All outer nodes	N	$N(\frac{N}{T}-1)$	$N(N\frac{T-1}{T}+T)$
Center node	T	N	0
Sum	$N+T$	$\frac{N^2}{T}$	$N(N\frac{T-1}{T}+T)$

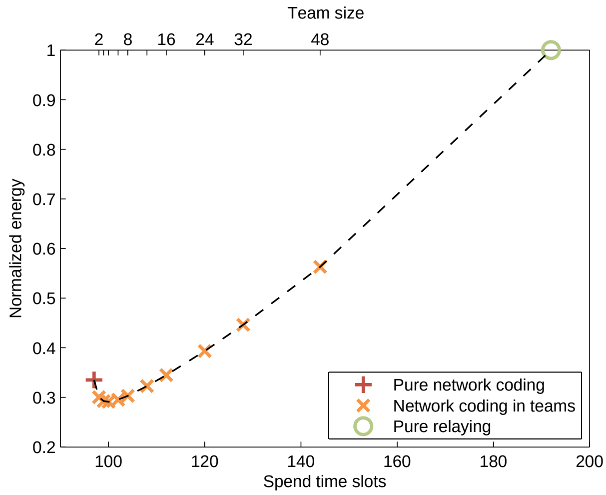
one team is active at a time, thus all nodes not in the active team are idle except for the center node which is always active. The operation in an active team is identical to that of pure network coding. Hence each of the N/T outer nodes in a team transmit its packet, which is received by the center node and all outer nodes in the team, except the sending nodes peer. The center node then combines all packets and transmit the coded packet. Thus, a total of $N/T+1$ transmissions is performed in each team. In this way each outer node receives a packet from the center node and from the N/T outer nodes in its team, except itself and its peer, thus $N/T-1$ packets. Each outer node is idle in the $T-1$ cases where its team is inactive and in one slot when its peer is transmitting, thus each outer node is idle in $(T-1)(N/T+1)+1$ slots, which can be rewritten to $N(T-1)/T+T$. The transmission time is given by the number of sending slots

$$S = N + T. \tag{3}$$

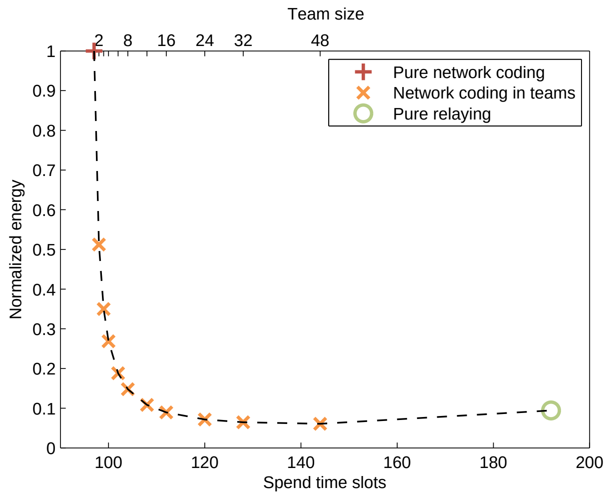
By the use of teams, higher transmission time can be traded for a lower number of receiving slots. Thus, the number of packets each of the outer nodes needs to overhear can be reduced by increasing T.

A.4 Performance Evaluation

We are now able to compare the performance of the three given schemes in terms of transmission time and energy in the wheel topology. We set the number of outer nodes to 96 as it allow us to test many different team sizes. Allowing many different team sizes provides important insight into how the different strategies changes operations of individual nodes and thereby affect energy consumption. We use the power consumptions given in Table 2. Based on this the energy consumption of all nodes in the cluster is calculated for pure relaying, pure network coding and network coding in teams of 2 to 48. These results are normalized and plotted against the transmission time of the scheme in Fig. 5(a). It can be seen that the energy consumption of pure network coding is lower than that of pure relaying. Interestingly the energy consumption is reduced when a small number of teams are used. However, if the number of teams becomes too large the energy consumption increases. The reduction in



(a)



(b)

Fig. 5. Energy consumption versus transmission time for pure relaying, pure network coding and network coding in teams: (a) When WLAN power consumption is assumed and (b) when an ideal technology power consumption ($P_{\text{idle}} = 0$) is assumed.

energy consumption comes at the price of a slightly increased transmission time.

To investigate how NC could perform in combination with a future wireless technology, we assume an ideal wireless technology where the idle power consumption is zero. Assuming such a technology is relevant because the idle power is relatively high for state of the art IEEE802.11 chipsets, however future chipsets may reduce this as energy consumption becomes increasingly important (e.g., driven by the use of VoIP on mobile devices). We use the same receiving and transmission power values as found for WLAN. Again we calculate the normalized power consumption and plot it against the transmission time, see Fig. 5(b). Note that the transmission time is the same as in the previous example. Because pure NC increases the time spent receiving considerably, while decreasing the idle time, it increases the amount of energy consumed significantly. When teams are used the energy consumption is reduced considerably, even below pure relaying, this however comes at the cost of increased transmission time compared to pure network coding.

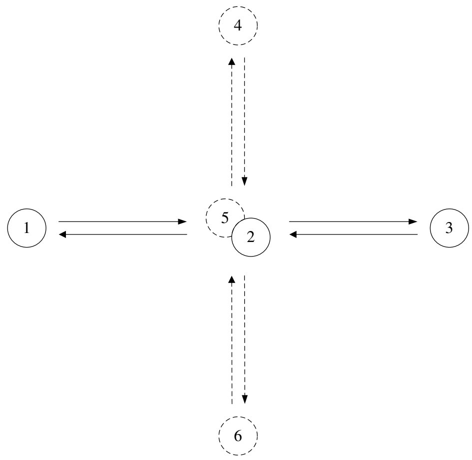


Fig. 6. The *wheel*++ example network.

These results demonstrate the advantages and disadvantages of pure relaying and network coding when deployed in WLAN. They also indicate that network coding in teams provide a good trade off between transmission time and energy consumption. The calculations for a future ideal technology shows that pure network coding can result in a significantly increased energy consumption if the idle power of the used wireless technology is very low.

B. An Expanded Wheel Topology

For the wheel scenario we showed that a team approach to NC is beneficial for the energy consumption. One scenario related to this approach, which is often used as an example in wireless cooperation, is a network consisting of a number of nodes holding disjoint data that is to be distribute among themselves [24]. In cooperative wireless networks, this technique can be used with e.g., multiple description coding (MDC) [25], where the use of cooperation can improve quality of service (QoS) while decreasing energy consumption. We combine this scenario with the wheel to form the *wheel*++ scenario illustrating cooperative wireless networks (CWN) combined with NC. The *wheel*++ consists of two clusters where each cluster contains three nodes that all cooperate to use the same service. Each node holds one packet which it wishes to transmit to the two other nodes in its cluster. The three nodes in a cluster is arranged on a line, thus the center node of both clusters are located in the middle of the wheel, as illustrated on Fig. 6. Both center nodes can receive and transmit packets to all other nodes. Each of the outer nodes is within transmission range of all other nodes, except the node in its cluster that is located diametrically opposite to itself. Thus, it should be noted that nodes can overhear transmissions of nodes in other clusters.

To illustrate the cooperative nature of the schemes presented for the wheel scenario we elaborate their operation in the *wheel*++ scenario. We use the same assumptions as in the wheel scenario and consider the same transmission schemes namely; pure relaying, pure network coding, and network coding in teams.

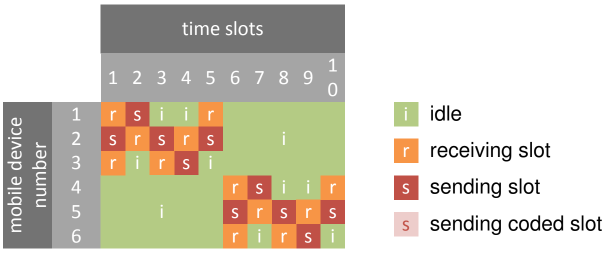


Fig. 7. Operation and number of actions performed in the wheel++ scenario when *pure relaying* is used.

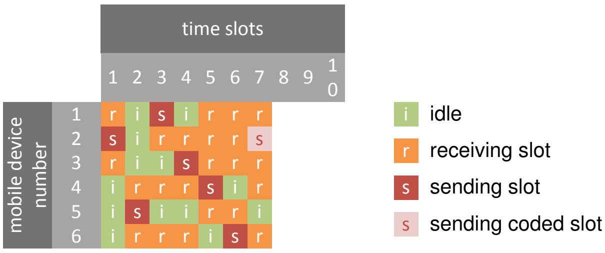


Fig. 8. Operation and number of actions performed in the wheel++ scenario when *pure network coding* is used.

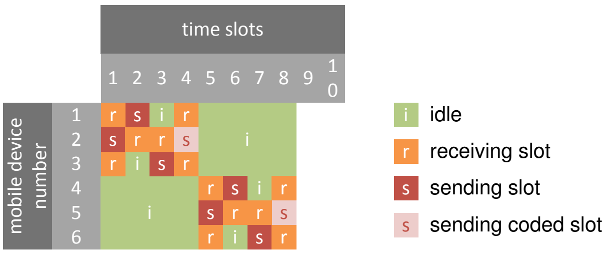


Fig. 9. Operation and number of actions performed in the wheel++ scenario when *network coding in teams* is used.

In *pure relaying*, to exchange the three packets, the center node broadcasts its own packet. The outer nodes in each cluster transmit their packet to the center node which forward these packets. While one cluster is active the other cluster is idle. The actions needed to performed the exchange is illustrated in Fig. 7.

In *pure network coding*, all center nodes broadcasts their packet. Each of the outer nodes transmit its packet which is received by one of the center nodes and all outer nodes except its peer. The center node which overhear all transmitted packets from the outer nodes, codes the packets from all outer nodes together and broadcast this. This packet is received by all outer nodes, that now holds $N-1$ packets and a combination of all N packets, which enables them to decode the packet they had not received.

In *network coding with teams*, the center nodes broadcasts their own packets followed by each outer node transmit its packet which is received by the center node of its cluster. The center node then codes the two received packets and broadcast the result. This enables the outer nodes to decode the packet they did not receive.

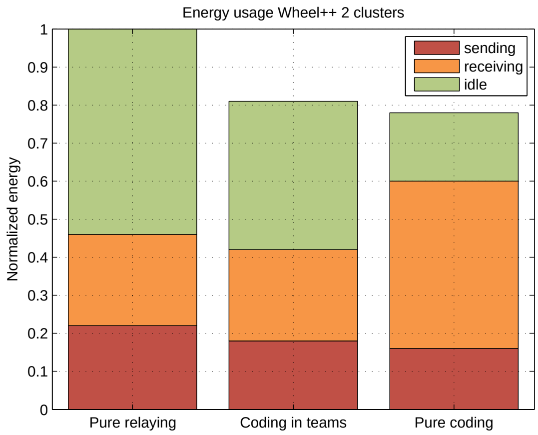


Fig. 10. Normalized energy consumption for pure relaying, network coding within a cluster, and pure NC for two clusters scenario.

B.1 Performance Evaluation

We calculate the energy consumption in order to determine how the different transmission strategies performs in the *wheel++* scenario. The energy consumption is calculated from the number of sending, receiving and idle slots and the power consumptions for WLAN given in Table 2. The ratio of idle and receiving slots for the different approaches can be seen by inspecting Figs. 7, 8, and 9. The normalized energy for pure relaying, pure network coding and network coding within a team, for the *wheel++* with six nodes is illustrated on Fig. 10. The energy consumption is lowest when pure network coding is used, followed by NC within a cluster and with pure relaying consuming the most energy. It should be noted that pure NC provides the best performance in terms of transmission time.

If we assume a more battery friendly technology where the idle power is considerably reduced compared to the receiving and/or sending power, in the ideal case $P_{idle} = 0$, the result is not surprisingly, very different. To obtain the resulting energy consumptions observe Figure 10 but ignore the top box for each of the bars, as this bar represent the idle consumption. With such a technology NC within a team will consume the least energy, pure relaying will consume slightly more and pure NC will consume the most energy, almost 30% more than NC in teams.

The results for the wheel and *wheel++* illustrates why *opportunistic listening*, advocated in [22] and [26], might perform poorly from an energy perspective in some scenarios, especially if future technologies minimizes the idle power consumption. Opportunistic listening is to overhear all surrounding traffic which can be used to create coding opportunities and thus create a more efficient packet transmission scheme. The authors demonstrate in a real life implementation that using opportunistic listening can increase the throughput. However the performance evaluation did not include the energy cost and thus increased energy consumption as a consequence of a continuously receiving wireless interface was not observed. Therefore, this approach may be problematic when it is combined with battery driven device and thus it should be used wisely on such devices. To conserve energy we should instead seek to minimize the use of the wireless interface instead of continuously listening, this poses an interesting problem which requires a fundamentally different approach to the use of NC in a wireless network.

IV. CONCLUSION

In this work, we have investigated several practical aspects and challenges arising when implementing network coding on real devices. This work provides insights into several different properties of NC that influence the energy consumption and thus is important when designing NC systems for battery driven devices. The performance of a basic RLNC implementation was evaluated on a high-end mobile phone. The results confirmed that the high computational complexity of RLNC create a performance bottleneck when compared to the achievable speed of the WLAN interface. During the measurements the high CPU utilization resulted in an increased power consumption. This problem should be addressed as power management and consumption is becoming an issue of increasing importance on mobile phones. To solve this problem an optimized implementation is needed, potentially using hardware accelerated methods. In addition, we have considered the communication in two simple topologies, this was done to investigate how network coding affects the operations performed by the individual devices. We found that pure network coding or opportunistic listening approaches as suggested by others, significantly change the ratio of the time a device spends in the idle and the receiving state. While pure network coding is beneficial in terms of transmission time, we saw that more coding lead to an increased energy consumption as the amount of devices increase. To mitigate this problem we have introduced the idea of forming network coding teams, which offer a trade off between transmission time and energy. The final performance analysis shows that the idle power of the WLAN technology has a huge impact on the potential energy savings.

This work has shown the potential of NC, but also its limitations when it is deployed on mobile platforms. For such platforms the use of NC should be carefully considered in order to achieve benefits in terms of transmission time without sacrificing the limited energy resource.

ACKNOWLEDGEMENTS

First of all we would like to thank Muriel Medard and Ralf Koetter for the discussion on network coding for the cooperative wireless networks. Furtheremore, we would like to thank Nokia for providing technical support as well as mobile phones to carry out the measurement campaign. Special thanks to Mika Kuulusa, Gerard Bosch, Harri Pennanen, Nina Tammelin, and Per Moeller from Nokia. This work was partially financed by the X3MP project granted by Danish Ministry of Science, Technology and Innovation.

REFERENCES

[1] T. Ho, R. Koetter, M. Medard, D. Karger, and M. ros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE ISIT*, 2003.

[2] "The network coding home page." [Online]. Available: http://hermes.lnt.e-technik.tu-muenchen.de/DokuWiki

[3] J.-S. Park, M. Gerla, D. S. Lun, Y. Yi, and M. Medard, "Codecast: A network-coding-based ad hoc multicast protocol," *Wireless Commun*, vol. 13, pp. 76–81, Oct. 2006.

[4] D. Nguyen, T. Nguyen, and B. Bose, "Wireless broadcasting using network coding," in *Proc. Third Workshop on Network Coding, Theory, and Applications*, Jan. 2007.

[5] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Proc. IEEE Conference on Information Sciences and Systems*, Mar. 2005.

[6] L. Popova, A. Schmidt, W. Gerstacker, and W. Koch, "Network coding assisted mobile-to-mobile file transfer," in *Proc. Australasian Telecommunication Networks and Applications Conference*, Dec. 2007.

[7] M. Wang and B. Li, "How practical is network coding?," in *Proc. IEEE IWQoS*, pp. 274–278, June 2006.

[8] M. Wang and B. Li, "Lava: A reality check of network coding in peer-to-peer live streaming," in *Proc. IEEE INFOCOM*, 2007, pp. 1082–1090.

[9] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 169–180, 2007.

[10] H. Shojania and B. Li, "Parallelized progressive network coding with hardware acceleration," in *Proc. IEEE IWQoS*, June 2007, pp. 47–55.

[11] M. Xiao and T. M. Aulin, "Energy-efficient network coding for the noisy channel network," in *Proc IEEE ISIT*, July 2006.

[12] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "A network coding approach to energy efficient broadcasting: From theory to practice," Tech. Rep., July 2005.

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

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, 2nd Ed.,*. The MIT Press, Sept. 2001.

[15] J. P. Vilela, L. Lima, and J. Barros, "Lightweight security for network coding," Tech. Rep., 2008.

[16] K. M. Greenan, E. L. Miller, and T. J. E. Schwarz, "Analysis and construction of Galois fields for efficient storage reliability," Tech. Rep., Aug. 2007.

[17] M. Hasan, "Look-up table-based large finite field multiplication in memory constrained cryptosystems," *IEEE Trans. Comput.*, vol. 49, pp. 749–758, July 2000.

[18] "Forum nokia." [Online]. Available: http://forum.nokia.com. Extensive resources about Nokia devices and developer related topics.

[19] N. Ravi, J. Scott, L. Han, and L. Iftode, "Context-aware battery management for mobile phones," in *Proc. IEEE International Conf. Pervasive Computing and Communications*, 2008, pp. 224–233.

[20] F. H. Fitzek, *Mobile Phone Programming: And its Application to Wireless Networking*, Chapter External Energy Consumption Measurements on Mobile Phones. Springer, Aug. 2007.

[21] G. B. Creus and M. Kuulusa, *Mobile Phone Programming: And its Application to Wireless Networking*, Chapter Optimizing Mobile Software with Built-in Power Profiling. Springer, Aug. 2007.

[22] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *Proc. SIGCOMM*, 2006, pp. 243–254.

[23] M. V. Pedersen, F. H. Fitzek, and T. Larsen, "Implementation and performance evaluation of network coding for cooperative mobile devices," in *Proc. IEEE Cognitive and Cooperative Wireless Networks Workshop*, May 2008.

[24] F. Fitzek and M. Katz, eds., *Cooperation in Wireless Networks: Principles and Applications – Real Egoistic Behavior is to Cooperate!* Springer, Apr. 2006.

[25] F. Albiero, J. Vehkaper, M. Katz, and F. Fitzek, "Overall performance assessment of energy-aware cooperative techniques exploiting multiple description and scalable video coding schemes," in *Proc CNSR*, Halifax, Nova Scotia, Canada, May 2008.

[26] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, "The importance of being opportunistic: Practical network coding for wireless environments," in *Proc. 43rd Allerton Conf. Communication, Control, and Computing*, 2005.

Janus Heide was born in Brønderslev, Denmark on March 20, 1982. He received his B.S in Electronics Engineering from Aalborg University, Denmark in 2007. He is currently pursuing a M.Sc. in wireless communication at Aalborg University. From July 2007, he has been working in the Mobile Devices research group at Aalborg University. His main research interests are protocol analysis and design, wireless networks, cooperative communication, network coding, and network performance evaluation.

Morten V. Pedersen was born in Sall, Denmark on September 20, 1982. He received his B.S in Electronics Engineering from Aalborg University in 2007. He is currently pursuing a M.Sc. in wireless communication. From January 2006, he has been working in the Mobile Devices research group at Aalborg University. His main research interests are programming, mobile networks, cooperative communication, network coding, and network performance evaluation.

Frank H.P. Fitzek is an Associate Professor in the department of Electronic Systems, University of Aalborg, Denmark heading the Future Vision group. He received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology - Rheinisch-Westflische Technische Hochschule (RWTH) - Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical University Berlin, Germany in 2002 for quality of service support in wireless CDMA networks. As a visiting student at the Arizona State University he conducted research in the field of video services over wireless networks. He co-founded the start-up company acticom GmbH in Berlin in 1999. In 2002, he was Adjunct Professor at the University of Ferrara, Italy giving lectures on wireless communications and conducting research on multi-hop networks. In 2005, he won the YRP award for the work on MIMO MDC and received the Young Elite Researcher Award of Denmark. He was selected to receive the NOKIA Champion Award in 2007 and 2008. In 2008, he was awarded the Nokia Achievement Award for his work on cooperative networks. His current research interests are in the areas of wireless and mobile communication networks, mobile phone programming, cross layer as well as energy efficient protocol design and cooperative networking.

Torben Larsen is a Full Professor in Aalborg University, Denmark. He received the Doctor Technices degree from Aalborg University in 1998. He has worked in various positions at Aalborg University, and at the companies Bosch Telecom and Siemens Mobile Phones. He is heading the Technology Platforms Section counting approx. 15 researchers working on technology platforms for wireless communication systems. He has published more than 90 peer reviewed journal and conference papers. He received the approx. 40,000 USD Spar Nord Research Prize in 1999 for excellent research work. In 2005, he was appointed by the Danish minister of research to be member of the research council for Technology and Production. In 2007, he was appointed Life Member of the Danish Academy of Technical Sciences. He has been leading several major research projects funded by EU, Danish and international companies and the Danish Government.