MBMS with user cooperation and network coding

Zhang, Qi; Heide, Janus; Pedersen, Morten Videbæk; Fitzek, Frank

Published in:
Globecom. I E E E Conference and Exhibition

DOI (link to publication from Publisher):
10.1109/GLOCOM.2011.6133643

Publication date:
2011

Document Version
Early version, also known as pre-print

User Cooperation with Network Coding for MBMS

Qi Zhang
Aarhus School of Engineering
Aarhus University
Email:qz@cs.au.dk

Janus Heide, Morten V. Pedersen and Frank H.P. Fitzek
Department of Communications Technology
Aalborg University
Email: [jah| mvp| ff]@es.aau.dk

***Abstract*—This paper applies user cooperation with network coding in MBMS (Multimedia broadcast/multicast service). MBMS uses Raptor codes which cannot overcome the tradeoff among bandwidth expansion, user perceived quality of service (QoS) and erasure correction performance. Therefore user cooperation together with network coding is proposed to support broadcast/multicast services in the future mobile communication networks to save bandwidth and to improve user perceived QoS without degrading erasure correction performance. The proposed approach is tailored to LTE networks being fully standard compliant. The simulation results show that local retransmissions can save up to 80% redundant information on the cellular link as long as there are at least two cooperative mobile devices. The results also show that network coding can save more than half of the traffic in the short-range link as long as there are four devices in the cooperation cluster.**

I. Introduction and Problem Statement

With the ever-increasing demand of diverse high bandwidth services on advanced mobile devices such as video streaming, software distribution, local news and weather reports and so on, the cellular network has high pressure in the limited available bandwidth. As such popular services are very often requested by several users, these download and streaming services can be distributed in broadcast/multicast mode. Multimedia broadcast/multicast service (MBMS) has been standardized in the Third Generation Partnership Project (3GPP) since 3GPP release 6. With Release 9 the Evolved MBMS ((e)MBMS) has been introduced for long term evolution (LTE) communication systems to support MBMS single frequency network (MBSFN). One of the most important challenges in (e)MBMS is error correction. As broadcast/multicast typically cannot reply on feedback specially for streaming services[1], ARQ/HARQ mechanism is not very suitable. Therefore, advanced forward error correction (FEC) technologies have to be employed in (e)MBMS to deal with this issue.

In the-state-of-art, FEC is applied at both physical layer and application layer in (e)MBMS. At physical layer, FEC is used to correct bit errors with the 3G air interface channel coding scheme. At application layer, Raptor codes are used to recover from lost packets.

Unfortunately, the current solution cannot overcome the tradeoff among *bandwidth expansion*, user perceived QoS and erasure correction performance. Therefore, this paper will advocate the usage of user cooperation combining with network coding to save bandwidth and to improve user perceived QoS without degrading erasure correction performance. The paper will introduce the-state-of-art approach with Raptor code and compare it with the proposed solution.

[1]Download services often requires the received data to be error-free in each mobile device. Therefore, it needs retransmission in the post-delivery phase to recover the undecodable packet erasures.

II. Raptor Code in (e)MBMS

Raptor codes provide packet-level protection at application layer to complement the bit-level FEC at physical layer. Raptor codes are implemented on both sides, i.e. the Broadcast Multicast Service Center (BMSC) and each individual mobile device. The basic encoding process is as following: A complete data file or a segment of a data stream is inserted into a large data block which is referred to as a *source block* (SB). One SB has k symbols and each symbol is composed of T bytes. Hence, one SB has kT bytes. After constructing a complete SB, the Raptor encoder generates $N - k$ repair symbols, each of size T. The number of repair symbols, i.e., $N - k$, depends on whether it is used for (e)MBMS streaming service or download service, the anticipated network conditions, the desired quality of delivery, the amount of available additional bandwidth or the allowed transmission time [1]. When the encoding is done, the BMSC sends the k source symbols followed by $N - k$ repair symbols to all the receivers, which is a *systematic code*. A systematic code has the advantage that received source symbols can be used by the receiver directly. Even in case of incomplete reception of information, the plain information in the first k symbols can be passed on to the higher layers. The receiver is able to decode the SB as long as it collects sufficient packets (no matter source or repair packets). If it does not receive sufficient packets to decode a SB, only the part of the SB that is directly received can be processed.

To choose the right SB size k and the number of repair symbols $N-k$, the tradeoffs among SB size, latency, expected quality of service, repair symbols and bandwidth expansion have to be taken into consideration. Let us take streaming service as an example. For streaming service, the *mean time between failures* (MTBF), denoted by τ here, is often used as a metric to measure the expected QoS. Say, each source block contains t_s seconds video content, the probability of decoding failure is P_{df}. The expected time between decoding failures will be t_s/P_{df} seconds. To meet the expected QoS, it is required that $t_s/P_{df} \geq \tau$. The probability of decoding

TABLE I
ASSUMED SETTINGS OF THE STREAMING SERVICE WITH RAPTOR CODES.

MTBF	Streaming Playout Speed	Symbol Size T	PER
3600s	512kbps	256 bytes	10%

TABLE II
OVERHEAD AND DELAY FOR DIFFERENT SOURCE BLOCK LENGTH

Source Block Length	1024	2048	4096	8192
Overhead	0.1238	0.1168	0.1136	0.1122
Delay (s)	4.0960	8.1920	16.3840	32.7680

failure is modeled in [9] by Equation 1.

$$P_{df} = \begin{cases} 1 & \text{if } \ m < k, \\ 0.85 \times 0.567^{m-k} & \text{if } \ m \geq k. \end{cases} \quad (1)$$

Whereby P_{df} denotes the decoding failure probability of the code with k source symbols if m symbols have been received [9]. For a given anticipated packet error rate p_e, assuming the BMSC sends N symbols in total for each source block and each symbol is encapsulated into one packet, then the number of the successfully received symbols can be expressed by $m = N(1 - p_e)$. The BMSC has to send at least N symbols for each block. N can be expressed by the following equation.

$$\begin{aligned} N &\geq \frac{1}{1-p_e}\left(\log_{0.567}\frac{t_s}{0.85\tau} + k\right) \\ &\geq \frac{1}{1-p_e}\left(k - 1.7624\ln\frac{t_s}{\tau} - 0.2864\right). \quad (2) \end{aligned}$$

Therefore, the minimum repair packets for each block, $R = N - k$, can be written as

$$R = \frac{1}{1-p_e}\left(p_e \cdot k - 1.7624\ln\frac{t_s}{\tau} - 0.2864\right). \quad (3)$$

Assume the data rate of video streaming is r bit/s, the perceived delay D can be expressed by $t_s = (8kT)/r$. The bigger source block size, the longer the perceived delay is. To ease the understanding here we present a small example with the settings listed in Table I. Table II gives the overhead and delay for different source block sizes under the settings. From Table II, we can see that the delay is linearly increasing with an increasing source block size. The overhead on the other side decreases with the source block size. Considering the perceived delay constraints, small block sizes are preferred. On the other side, the small block sizes will need more repair symbols to have the same performance as large block size.

Furthermore, the overhead presented in Table II is the ideal theoretical value under condition of the perfect prediction of the packet erasure rate. In reality the packet erasure pattern among all the mobile devices in wireless multicast network varies dramatically over time. The reason is that packet erasure can be induced by a variety of causes, for instance, network congestion, deep fading, severe path loss, interference, hardware performance and many more. Especially the hardware performance is often underestimated, which has been proved by many real measurements [10], [11], [12]. Therefore, the packet erasure rate within two block durations can be very different, especially in the small block size case.

III. USER COOPERATION FOR ERASURE RECOVERY

The concept of user cooperation was proposed in [4], [13]. The basic idea of user cooperation is that a mobile device can exchange packets with the neighbor devices in its proximity over short-range links to achieve a common or individual goal. The generic network architecture of user cooperation is shown in Fig. 1. Since the mobile devices in shelf are all multi-mode devices, namely they do not only have cellular interface but also short-range link interface, such as WiFi, the user cooperation can be implemented in commercial mobile devices. It has been proved to be a promising scheme to tackle many tradeoffs that the cellular network itself cannot overcome. Many research works, such as [6], [8], [14], [15], have shown the gain of user cooperation in energy saving, throughput enhancement, etc. In this work user cooperation is applied for erasure recovery in (e)MBMS. It can reduce the overhead introduced by repair symbols. Furthermore, it can reduce the perceived delay by using small source block meanwhile without degrading the Raptor code error correction performance.

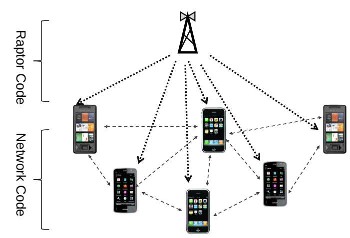

Fig. 1. Network architecture of user cooperation.

Though the packet erasure pattern is a combination of many causes in multicast wireless network, the independent packet erasure among mobile devices is dominant in all the lost packets. It means that the neighbor devices of a mobile device often have the packets that this mobile device has lost. In other words, those devices can use user cooperation to help each other to recover the lost packets. In (e)MBMS it takes a few seconds for the sender to send out a batch of encoded packets in each block. Likewise, it will take the receiver a few seconds to receive enough packets to construct the decoding block. In the proposed cooperative erasure recovery scheme, it is not necessary for the mobile devices to start erasure recovery until the end of a block. The mobile devices can progressively recover the erasures locally on the run, for instance performing local recovery every 64 packets, which does not only reduce the number of the needed repair symbols from base station, but also highly saves the time to receive a large amount of repair symbols and the time to decode in the end. The illustration of progressive recovery in user cooperation is shown in Fig. 2.

In local retransmission with user cooperation, the set of the

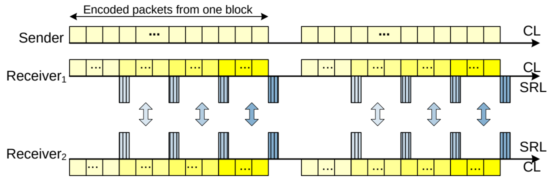

Fig. 2. Illustration of progressive recovery in user cooperation.

all packet erasures of a cluster can be expressed by

$$\hat{L} = L_1 \cup L_2 \ldots \cup L_n \,, \tag{4}$$

where $L_1, L_2, \ldots, L_n$ are the sets of the packet erasures of mobile device 1, 2,..., n, respectively.

The common packet erasures in the cluster can be written as

$$\Lambda = L_1 \cap L_2 \ldots \cap L_n . \tag{5}$$

When $\Lambda = \phi$, it means that all the packet erasures can be recovered within the cluster. The total number of the recoverable packets by cooperative retransmission, N_{rv}, is given by

$$N_{rv} = |\hat{L}| - |\Lambda| \,, \tag{6}$$

where $|\hat{L}|$ is the cardinality of set $\hat{L}$, i.e. the number of the total packet erasures among the mobile devices. $|\Lambda|$ is the number of the packet erasures of the cluster, i.e. the correlated packet erasures of the mobile devices.

With local retransmission, BMSC can send N_{coop} symbols for each source block, which includes source symbols and repair symbols. Then the number of the successfully received symbols, m can be expressed as

$$m_{coop} = N_{coop} - |\Lambda|. \tag{7}$$

The minimum number of symbols should be sent by BMSC can be obtained from the following equation.

$$N_{coop} \geq k + |\Lambda| - 1.7624 \ln \frac{t_s}{\tau} - 0.2864 \;. \tag{8}$$

With user cooperation, the minimum repair symbols for each source block, R_{coop}, becomes

$$R_{coop} = N_{coop} - k = |\Lambda| - 1.7624 \ln \frac{t_s}{\tau} - 0.2864 \;. \tag{9}$$

In the cellular link, user cooperation based local retransmission can reduce the minimum required repair symbols for each block from R to R_{coop}. Therefore, the user cooperation gain, G_{coop}, can be written as following

$$G_{coop} = \frac{R - R_{coop}}{R} = 1 - \frac{(1 - p_e^c)(|\Lambda| + \xi)}{p_e^c \cdot k + \xi} \tag{10}$$

where, $\xi = -1.7624 \ln \frac{t_s}{\tau} - 0.2864$ and p_e^c is the cellular link packet erasure rate of the worst mobile device in a multicast group.

Comparing the case of cooperation with the case of no user cooperation, it can clearly see from Equation 10 that the cooperation gain depends on the correlated packet erasures $|\Lambda|$. The smaller $|\Lambda|$ is, the larger the cooperation gain will be. Assuming source block length k equal to 1024 and the rest setting as in Table I, it can have 82.72% cooperation gain when $|\Lambda|$ is 10% of the total erasures $|\hat{L}|$. G_{coop} reaches its upper bound when $|\Lambda|$ equals to zero. The maximum cooperation gain is 90.8%. Note that here 90.8% is a theoretical value assuming that the real packet erasure rate is equal to the average packet erasure rate. In practise the real packet erasure rate could be larger than the average. Therefore, the maximum cooperation gain will be slightly hihger than 90.8%.

TABLE III
COOPERATION GAIN

$\|\Lambda\|$	0	$0.1\|\hat{L}\|$	$0.2\|\hat{L}\|$	$0.3\|\hat{L}\|$	$0.4\|\hat{L}\|$
G_{coop}	0.9080	0.8272	0.7464	0.6656	0.5848

Furthermore, the local retransmission with user cooperation can save a large amount of the retransmission traffic in the post-delivery phase for download services, as most of the erasures in the cluster can be recovered over the short-range link. Additionally, it takes less time for each mobile device to recover all the erasures, as the data rate of the short-range link is usually higher than that of cellular link.

The cooperation gain achieved in cellular links is very impressive, at the meanwhile, it has some costs over the short-range link. The cost including two parts: the communication cost of mobile devices and the network resource cost.

The communication cost of mobile devices is mainly the energy used in short-range communication. As the energy per bit ratio (EpBR) of short-range link is much lower than that of cellular link, there is still energy saving gain in the overall energy consumption of the mobile devices even though user cooperation has energy cost on the short-range link, which has been proved in many works [14], [15], [6], [8].

To evaluate the network resource cost, it is necessary to calculate the number of exchanged packets over short-range link, N_{sr}. It can be expressed as

$$N_{sr} = \frac{N_{rv}}{1 - p_e^{sr}} = \frac{|\hat{L}| - |\Lambda|}{1 - p_e^{sr}} \,, \tag{11}$$

where, p_e^{sr} is the packet erasure rate in the short-range link. In reality, p_e^{sr} is smaller than p_e^c.

On the one hand, as short-range link usually uses the license-free spectrum and has much higher data rate than that of cellular link, the network resource used for packet exchange over short-range link is ideally regarded as free. On the other hand, considering more and more applications starting to exploit the lower cost short-range link, there is still need to use the short-range network resource in an efficient way. The next section will address how to more efficiently exchange the packets in the short-range link.

IV. Network Coding Applied in User Cooperation

Network coding was first introduced by [16] which showed the achievable multicast capacity by network coding mixing the information from different flows. [17] proved that linear coding can obtain the multicast capacity bound. Furthermore, [18] shows that linear coding with random coefficients can be used to reach the capacity bound. Therefore, random linear coding is widely applied in network coding. In contrast to end-to-end erasure code such as Raptor code, the fundamental characteristics of network coding is that network coding introduces additional encoding processes at the intermediate nodes. In a nutshell, network coding is not a specific coding scheme but a novel transmission scheme combined with a proper coding scheme such as random linear coding.

In random linear network coding source data is divided into symbols of length T. The number of original symbols over which encoding is performed is referred to as the batch size or generation size, denoted by g. Thus the g original symbols of length T are arranged in the matrix $\mathbf{M} = [\mathbf{m}_1\mathbf{m}_2 \ldots \mathbf{m}_g]$, where $\mathbf{m}_i$ is a column vector. Additionally all operations are performed over a Galois field of size q [19].

To encode a symbol $\mathbf{x}$ at the source, $\mathbf{M}$ is multiplied with a randomly generated vector $\mathbf{g}$ of length g, $\mathbf{x} = \mathbf{M} \times \mathbf{g}$. In this way we can construct $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_{g+r}]$ that consists of $g + r$ coded data symbols and $\mathbf{G} = [\mathbf{g}_1\mathbf{g}_2 \ldots \mathbf{g}_{g+r}]$ that contains $g + r$ randomly generated encoding vectors, where r is the number of redundant symbols.

Furthermore, any relay or sink node that has received $g' > 1$ linear independent symbols, can recode and thus create new coded symbols ($g' \leq g$). When a sink has received g linear independent coded packets and encoding vectors, it can successfully decode the original data packets. All received coded packets are placed in the matrix $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1\hat{\mathbf{x}}_2 \ldots \hat{\mathbf{x}}_g]$ and all encoding vectors are placed in the matrix $\hat{\mathbf{G}} = [\hat{\mathbf{g}}_1\hat{\mathbf{g}}_2 \ldots \hat{\mathbf{g}}_g]$. The original data $\mathbf{M}$ can then be decoded as $\hat{\mathbf{M}} = \hat{\mathbf{X}} \times \hat{\mathbf{G}}^{-1}$.

The main motivation of applying network coding (NC) to user cooperation is to improve the cooperation efficiency for the short-range link. First of all the packet exchange among cooperative mobile devices is reduced to a minimum by NC. In other words, NC can use less than $|\hat{L}|-|\Lambda|$ exchanged packets to correct all the recoverable erasures in the cooperative cluster. The reason lies in that an encoded packet containing information of multiple packets, many nodes that have different erasures can benefit from the same encoded packet. Furthermore, considering the partially connected cooperative cluster, i.e. the clusters where peers cannot communicate directly with each other, but information are relayed within the cluster, the recoding characteristics of NC can make the packet exchange very efficient. Therefore, NC will help to reduce the overall traffic and energy used for cooperation in the short-range link. Note that NC only has impact on the short-range link and it keeps the performance of cellular link the same as that of user cooperation without NC. In the following, we are going to analyze and derive the number of exchanged coded packets in the short-range link.

Assuming mobile device i has the most of packet erasures, and mobile device j has the least of packet erasures, there are

$$|L_i| = max\{|L_1|, |L_2|, \ldots, |L_n|\} \quad (12)$$

$$|L_j| = min\{|L_1|, |L_2|, \ldots, |L_n|\} \; . \quad (13)$$

In the case that mobile device i and j do not have any common erasures, mobile device j can send coded packets to repair all the recoverable erasures in mobile device i. These coded packets can also be used to correct the erasures at other mobile devices in the cluster. Though they have different erasures, as long as the number of the recoverable erasures of the other mobile devices is less than that of mobile device i, all the other mobile devices can recover their erasures by overhearing the coded packets. When mobile device i gets its erasures corrected, it can send coded packets to correct the erasures of mobile device j. In this case the number of the exchanged coded packets can be expressed as

$$N_{nc} = (|L_i|-|\Lambda|) + (|L_j|-|\Lambda|) \quad (14)$$

where, $|\Lambda|$ is the number of the packet erasures of the cluster, i.e. the correlated packet erasures of the mobile devices which are not recoverable by local retransmission.

In the case that mobile device i and j have some common erasures besides the ones in Λ, the rest of nodes can help to correct these erasures. In this case N_{nc} is less than that of the former case, i.e.

$$N_{nc} < (|L_i|-|\Lambda|) + (|L_j|-|\Lambda|) \, , \quad (15)$$

which gives an upper bound of the exchanged coded packet in the short-range link.

Next, we look at the issue from another angle. The set of packets only received by mobile device k is denoted Δ_k. There is

$$\Delta_k = \Lambda_{-k} \setminus \Lambda \, , \quad (16)$$

where, $\Lambda_{-k} = L_1 \cap L_2 \ldots L_{k-1} \cap L_{k+1} \ldots \cap L_n$.

To correct all the recoverable erasures at mobile device k it must receive $|L_k| - |\Lambda|$ packets. Mobile device k must also send $|\Delta_k|$ packets, as it is the only one that holds these packets. Therefore, the number of the exchanged packets that node k involved is expressed by $|\Omega_k|$,

$$|\Omega_k| = (|L_k| - |\Lambda|) + |\Delta_k| \, . \quad (17)$$

If $|L_i| \leq max\{|\Omega_1|, |\Omega_2|, ..., |\Omega_n|\}$, then let us assume only mobile device k has these packets Δ_k and $|\Omega_k|$ is the largest among $\{|\Omega_1|, |\Omega_2|, ..., |\Omega_n|\}$. After mobile device k has exchanged $|\Omega_k|$ encoded packets by network coding in the cluster, this procedure does not only help mobile device i to recover the missed packets, but also distributes the unique $|\Delta_k|$ packets of mobile device k among the cluster. Therefore,

$$N_{nc} = max\{|\Omega_1|, |\Omega_2|, ..., |\Omega_n|\} \; . \quad (18)$$

If $|L_i| > max\{|\Omega_1|, |\Omega_2|, ..., |\Omega_n|\}$, then the exchanged encoded packets among the cluster must be at least $|L_i|$. Therefore,

$$N_{nc} > max\{|\Omega_1|, |\Omega_2|, ..., |\Omega_n|\} \; , \quad (19)$$

which gives a lower bound on the number of the exchanged packets in the short-range link.

Thus we obtain the range of N_{nc} as

$$max\{|\Omega_1|, |\Omega_2|, ..., |\Omega_n|\} \leq N_{nc} \leq |L_i| + |L_j| - 2|\Lambda| \, . \quad (20)$$

The above derived N_{nc} is an ideal number. To be precise, the linear independent probability of the received coded packets and packet erasure rate of short-range link should be taken into account.

Given that the sender holds g linear independent symbols, the probability that a received coded symbol is linearly independent is given in [20] by.

$$P_{ind} = 1 - \frac{1}{q^{g-g'}} , \quad (21)$$

where g' is the number of the received independent symbols at the sink. Thus the number of exchanged coded symbols that must be received can be calculated as

$$E[N] = \sum_{g'=0}^{g-1} \left(1 - \frac{1}{q^{g-g'}}\right)^{-1} . \quad (22)$$

Hence, the number of the exchanged coded packets to correct all the recoverable erasures locally is expressed by

$$N_{sr}^{nc} = \frac{N_{nc} \cdot E[N]}{1 - p_e^{sr}} , \quad (23)$$

when q is high $E[N] \approx 1$, thus,

$$N_{sr}^{nc} \approx \frac{N_{nc}}{1 - p_e^{sr}} . \quad (24)$$

The value of N_{nc} in practice depends on the local retransmission scheme with network coding. In other words, if the local retransmission scheme is designed well N_{nc} will be close to the lower bound. The local retransmission can be implemented in many possible ways. We propose one as an example here. The basic idea is that the *current* "best" mobile device, i.e., the one with the least packet erasures, first sends an encoded packet with all the packets it has received. The benefit of such an encoded packet is twofold. On the one hand, the encoded packet has the highest probability to correct the most erasures in the other mobile devices. On the other hand, it implicitly indicates what packets it misses. The missed packets of the *current* "best" mobile device are regarded as the current *rare* packets. Note that a mobile device defers a certain period according to its back-off timer before it sends the coded packets. The value of the timer is a function of the number of packets and the number of rare packets the mobile device has. The more (*rare*) packets a mobile device has, the less the back-off time is. Thus by receiving the encoded packets, some "non-best" mobile devices become better. Then one of these mobile devices will become the *current* "best" mobile device. It will start sending encoded packets until it is replaced by another "better" one. As soon as the *current* "best" mobile device sends out an encoded packet which includes all the packets of this generation, all of the others reset their back-off timer. Then the back-off timer is used for sending a feedback, the value of which is a function of the number of packets a mobile device needs. The more packets a mobile device still needs, the less back-off time is. Thus the current "worst" mobile device can give a short feedback to indicate how many packets it still needs. Then the *current* "best" mobile device will stop sending after sending out the needed number of packets.

V. Simulation Results

In this section the two main simulation results, the gain of local retransmission with user cooperation and the gain of network coding applied to the short-range link, are presented. According to the analysis in Section II, we know that the erasure correction performance depends on the source block size. The smaller the source block size is, the more overhead it carries. 3GPP limits the minimum source block size to 1024 due to the inefficiency of smaller block size than 1024, though smaller block sizes has shorter user perceived latency. With user cooperation it can further reduce the block size to 512 or even 256. Assume that the average packet erasure rate is 10%. Fig. 3 shows the simulation results of user cooperation. When there are two, three and four mobile devices in the cooperation cluster and the block size is 1024, it can save 80.6%, 89.0% and 92.4% overhead in the cellular link, respectively. It clearly shows the significant overhead saving by user cooperation. Furthermore, as long as there are two devices in the cooperation cluster, the overhead of block size equal to 512 and 256 is only 6% and 10%, respectively. Such overhead is lower than the overhead of the case that block size is equal to 1024 and no cooperation is involved. It means that it is feasible to use smaller block size such as 512 and 256 with two user cooperation. It is also obvious that using block size 512 and 256, the perceived delay can be reduced by 50% and 75%, respectively. To sum up, in the state of the art system overhead can only be reduced by scarifying latency with large block size and vice versa. User cooperation on the other side offers both, low latency and low overhead.

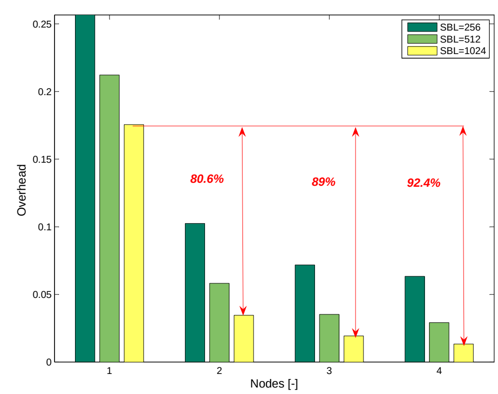

Fig. 3. Overhead saving in the cellular link by user cooperation.

To show the main benefit of network coding, we compare the number of the exchanged packets to recover all the erasures in user cooperation with and without network coding. The generation size of network coding is assumed 64 here. The packet erasure rate of cellular link and short-rang link is 10% and 5%, respectively. The simulation result is shown in Fig. 4. Network coding starts to work when there are more than two cooperation devices. It shows with network coding cooperation needs much fewer packets to be exchanged. For instance, network coding heuristic approach saves more than

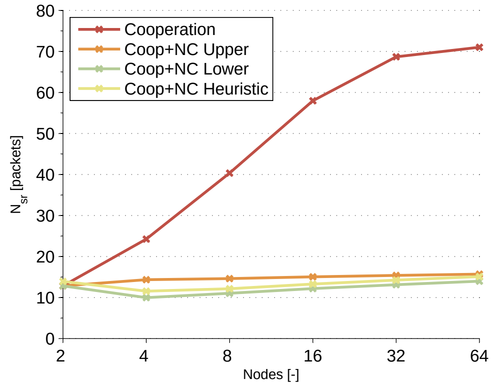

Fig. 4. Comparison of the number of the exchanged packets over the short-range link with/without network coding.

half and 75% of the exchanged packets when there are four and eight devices in the cooperative cluster, respectively. Furthermore, the figure shows that the number of exchanged packets increases only smoothly with network coding with an increasing number of cooperating users. However, it increases dramatically without network coding. Additionally, it shows that the performance of the proposed local retransmission scheme with heuristic network coding approach is very close to the derived lower bound.

VI. Conclusion

To tackle the drawbacks of Raptor code in (e)MBMS, a novel local retransmission scheme based on the concept of user cooperation has been proposed in addition to the usage of Raptor coding. The simulation results show that local retransmission can save about 80% overhead in the cellular link as long as two mobile devices cooperate. Larger gains can be achieved by increasing the number of cooperating devices. Furthermore, local retransmission makes it feasible to use smaller block sizes on the cellular link using Raptor codes to reduce the user perceived delay and to improve user perceived experience. To make the local retransmission in the short-range link more efficient, network coding is considered for the local retransmission and a first local retransmission protocol is proposed. The simulation results show that network coding can save more than half of the short-range traffic as long as there are four mobile devices in the cooperation cluster. Reducing the traffic on the short range link will reduce the overall energy consumption as well as it will reduce the time that is need to complete the exchange of local packets especially in the dense traffic networks.

VII. Acknowledgments

This work was partially financed by the CONE project (Grant No. 09-066549/FTP) granted by Danish Ministry of Science, Technology and Innovation as well as by the collaboration with NOKIA throughout the ENOC project.

References

[1] How DF Raptor is Used in MBMS. Technical report, Digital Fountain, March 2007.
[2] M. Luby. LT Codes. In *43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS),*, pages 271–282, Vancouver, Canada, Nov 2002.
[3] A. Shokrollahi. Raptor Codes. *IEEE transactions on information theory*, 52(6):2551–2567, June 2006.
[4] Frank H.P. Fitzek and Marcos D.Katz, editors. *Cooperation in Wireless Networks: Principle and Applications*. ISBN-10 1-4020-4710-X. Springer, 2006.
[5] Frank H.P. Fitzek and Katz Marcos, editors. *Cognitive Wireless Networks: Concepts, Methodologies and Visions*. ISBN: 978-1-4020-5978-0. Springer, 2007.
[6] Q. Zhang, F.H.P. Fitzek, and Marcos Katz. Cooperative Power Saving Strategies for IP-Services Supported over DVB-H Networks. In *Wireless communication&network conference*, March 2007.
[7] Qi Zhang, Frank H.P. Fitzek, and Villy B. Iversen. Design and Performance Evaluation of Cooperative Retransmission Scheme for Reliable Multicast Services in Cellular Controlled P2P Networks. In *The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC07)*, September 2007.
[8] Qi Zhang, Frank H.P. Fitzek, and V. B. Iversen. One4all Cooperative Media Access Strategy in Infrastructure Based Distributed Wireless Networks. In *IEEE Wireless Communication and Networking Conference (WCNC)*, April 2008.
[9] M. Luby, M. Watson, T. Gasiba, and t. Stockhammer. Mobile Data Broadcasting over MBMS Tradeoffs in Forward Error Correction. In *the 5th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, Stanford, CA, USA, December 2006.
[10] P. Fuxjager and F. Ricciato. Collecting broken frames: Error statistics in ieee 802.11b/g links. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pages 30–35, April 2008.
[11] Janus Heide, Morten V. Pedersen, Frank H.P. Fitzek, Tatiana V. Kozlova, and Torben Larsen. Know Your Neighbour: Packet Loss Correlation In IEEE 802.11b/g Multicast. In *4th International Mobile Multimedia Communications Conference (MobiMedia)*, Oulu, Finland, 7-9 July 2008.
[12] Hoi-Sheung Wilson So, Kevin Fall, and Jean Walrand. Packet Loss Behavior in a Wireless Broadcast Sensor Network.
[13] F.H.P. Fitzek, M. Katz, and Q. Zhang. Cellular Controlled Short-Range Communication for Cooperative P2P Networking. In *Wireless World Research Forum (WWRF) 17*, volume WG 5, Heidelberg, Germany, November 2006. WWRF.
[14] F.H.P. Fitzek, M. Katz, and Q. Zhang. Cellular Controlled Short-range Communication for Cooperative P2P networking. *Wireless Personal Communications*, 2008.
[15] Madsen T K., Zhang Q., and Fitzek F H P. Design and Evaluation of IP Header Compression for Cellular-Controlled P2P Networks. In *IEEE International Conference on Communication (ICC)*, June 2007.
[16] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.
[17] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear Network Coding. *IEEE Transactions on Information Theory*, 49(2):371–381, February 2003.
[18] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Transactions Information Theory*, 52(10):4413–4430, Oct. 2006.
[19] C. Fragouli, J.L. Boudec, and J. Widmer. Network Coding: an Instant Primer. *SIGCOMM Comput. Commun. Rev.*, 36(1):63–68, 2006.
[20] A. Eryilmaz, A. Ozdaglar, and M. Medard. On Delay Performance Gains From Network Coding. *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 864–870, March 2006.
[21] Q. Zhang, Frank H. P. Fitzek, and Marcos Katz. *WiMAX Evolution: Emerging Technologies and Applications*, chapter Cooperative Principles in WiMAX. John Wiley & Sons. Ltd, 2008.