Network Coding Based Cooperative Peer-to-Peer Repair in Wireless Ad-Hoc Networks

Xin Liu, Saqib Raza, Chen-Nee Chuah
University of California, Davis
Emails:{xyzliu, sraza, chuah}@ucdavis.edu

Gene Cheung
Hewlett-Packard Laboratories Japan
Email: gene-cs.cheung@hp.com

Abstract— **Cooperative Peer-to-Peer Repair (CPR) has been proposed to recover from packet losses incurred during 3G broadcast. CPR leverages the increasing presence of multi-homed mobile devices having both 3G cellular and IEEE 802.11 wireless interfaces. Mobile devices can, therefore, draw upon IEEE 802.11 peering links to cooperatively achieve out-of-band repair of 3G broadcasting losses. This paper considers the problem of employing Network Coding (NC) to exploit the broadcast nature of the wireless medium towards enhancing the efficiency of CPR. We show that the minimum latency scheduling problem for NC based CPR (NC-CPR) is NP-Hard. We present heuristics for NC-CPR that assume a priori topology and packet loss information. Insights gained from our heuristics are leveraged to propose NC-DCPR, a fully distributed protocol for NC-CPR. We conduct extensive simulation experiments under realistic network conditions. Our results show that employing network coding significantly improves the efficiency of CPR.**

I. INTRODUCTION

Multimedia Broadcast/Multicast Service (MBMS) [1] offers rich content distribution, such as video, audio streaming, and file sharing, in UMTS cellular networks of 3GPP version 6.0 or later. Due to the unpredictable and time-varying nature of wireless networks, packet losses are inevitable even when complex error-correction channel coding schemes are implemented in the system. Studies in [2] and [3] observe that the widespread availability of multi-homed mobile devices [4] having both 3G cellular and IEEE 802.11 wireless interfaces can be leveraged to perform *out-of-band* repair of 3G broadcasting losses. In such Cooperative Peer-to-Peer Repair (CPR), nodes simultaneously receiving identical MBMS content can collaboratively repair lost packets over IEEE 802.11 links. In other words, nodes that possess certain packets can send them to nodes that failed to receive them. CPR is similar in spirit to the popular Internet content delivery application BitTorrent where users help each other to download commonly desired files. Our previous work [2] outlines a distributed CPR protocol, DCPR, that successfully recovers all lost packets at all network nodes within a reasonable amount of time.

A key feature of DCPR is *batching*, wherein network nodes wait for a batch of broadcast packets before triggering the repair process. This allows the cost associated with disseminating control information, required by distributed nodes to schedule CPR transmissions, to be amortized over multiple packets [2]. Repairing packets in batches also makes it possible to employ Network Coding (NC), which has been shown that by exploiting the shared nature of wireless broadcast, one can
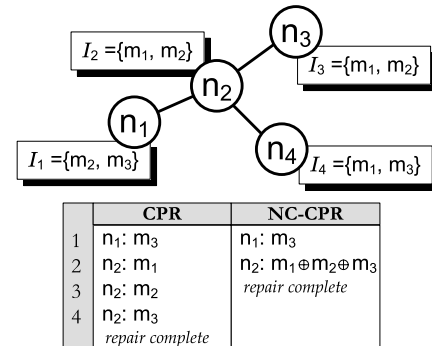

Fig. 1.   Advantage of Network Coding for Cooperative Peer-to-Peer Repair

increase network throughput [5], [6]. This paper addresses the NC-CPR problem, which looks at how to employ NC to enhance the efficiency of CPR. We illustrate our motivation with the help of a simple example. Fig. 1 shows four nodes, with arcs between node-pairs implying that the nodes are within transmission range of each other. The batch of packets being repaired comprises of three packets m_1, m_2, and m_3. The set $\mathcal{I}_i$ gives the packets initially present at a node n_i. It can be seen that n_2 needs to receive m_3. Furthermore, we require three additional transmission rounds for n_2 to transmit packets m_1, m_2, and m_3 to nodes n_1, n_4, and n_3, respectively. Successfully repairing all missing packets requires at least four transmission rounds if nodes employ the traditional store-and-forward paradigm. However, allowing network nodes to encode packets before transmitting them can reduce the time required to repair all packets. Fig. 1 shows how a simple XOR-based coding scheme allows us to repair all missing packets in just two transmission rounds.

Our contributions in this paper are threefold:

- We show that finding the optimal scheduling for the minimum latency NC-CPR problem is NP-hard, which means there is no polynomial-time algorithm of the size of the input instance that can discover the optimal schedule unless P=NP. We define a lower bound for NC-CPR and use it as a performance benchmark.
- We design a heuristic algorithm NC-CCPR for centralized NC-CPR. NC-CCPR assumes global knowledge of network topology and packet losses to compute a schedule of CPR transmissions.

- Based on the insights gained from NC-CCPR, we propose NC-DCPR, a fully distributed NC-CPR protocol. NC-DCPR is a network coding based solution for CPR which addresses practical concerns. Extensive simulation studies show the effectiveness of our protocol under a host of network settings.

Many pervious studies have explored message dissemination using NC. In [7], nodes perform random NC when transmitting packets and each transmitted packet is a random linear combination of the original messages. We leverage this idea to design the NC-CPR protocol. A gossip-based protocol is proposed in [8] which utilizes network coding to disseminate messages to all nodes in time $O(n)$ where n is the number of nodes in the network. Instead of gossiping, we utilize the broadcast nature of the wireless medium to disseminate packets. In [9], the authors consider the problem of minimizing total broadcast energy, which translates to minimizing total number of transmissions, while in this paper we consider minimizing the total repair time instead.

The paper is organized as follows: In Section II, we present the network model and formalize the NC-CPR scheduling problem as a combinatorial optimization problem. We show NC-CPR to be NP-Hard in Section III. We also propose a centralized heuristic algorithm and the NC-DCPR in Section III. Section IV evaluates the performance of NC-DCPR through extensive simulation studies.

II. PRELIMINARIES

We consider a wireless network in which all nodes are equipped with two wireless access interfaces, i.e., 3G cellular and IEEE 802.11 wireless interfaces. All nodes operate in IEEE 802.11 broadcast mode utilizing the wireless broadcast advantage: each node's transmission is overheard by nodes in its neighborhood. Nodes wait for a batch of broadcast packets to be delivered through MBMS before triggering the repair process to cooperatively recover lost packets using their IEEE 802.11 interfaces. Lost packets should be recovered before the arrival of the next batch of packets, whereupon the repair process for the new batch is triggered.

A node can successfully receive a packet if and only if one of its neighboring nodes transmits the packet, and the node is not within interference range of any other transmitting node. We model the CPR network as a directed graph $G = (\mathcal{N}, \mathcal{L}_\mathcal{T}, \mathcal{L}_\mathcal{I})$, where $\mathcal{N}$ is the node set of size N, $\mathcal{L}_\mathcal{T}$ is the transmission link set, and $\mathcal{L}_\mathcal{I}$ is the interference link set. We have a set of packets $\mathcal{M}$ of size M, and each packet is B bytes in length. At the onset of the repair process for a batch, each node $n_i \in \mathcal{N}$ has a set of packets $\mathcal{I}_i \subseteq \mathcal{M}$. As in [2], we assume that for every packet $m \in \mathcal{M}$, there exists at least one node $n_i \in \mathcal{N}$, such that $m \in \mathcal{I}_i$.

The repairing process is composed of discrete transmission rounds. A subset of nodes are selected to transmit in each round. The entire repair process can be characterized by *packet transmission*, *packet reception*, and *transmission scheduling*.

a) Packet Transmission: We use $\mathcal{U}_i^t$ to denote the set of packets present at n_i at round t. Initially, $\mathcal{U}_i^0 = \mathcal{I}_i$. A packet P_i^t, transmitted by n_i during transmission round t, is a random linear combination of packets in $\mathcal{U}_i^t$. Formally, $P_i^t = \sum_{k=1}^{K_i^t} c_{i,k}^t * p_{i,k}^t$, where $K_i^t = |\mathcal{U}_i^t|$, $c_{i,k}^t$ is the randomly generated coefficient from a finite field GF (256), and $p_{i,k}^t \in \mathcal{U}_i^t$. $p_{i,k}^t$ is itself a random linear combination of all packets in $\mathcal{M}$ with coefficient vector $\overrightarrow{V}_{i,k}^t$ of length M.

We denote $\overrightarrow{E}_i^t = \sum_{k=1}^{K_i^t} c_{i,k}^t * \overrightarrow{V}_{i,k}^t$ as the M vector to be transmitted with the encoded packet P_i^t. P_i^t is a linear combination of all packets in $\mathcal{M}$ with coefficient vector $\overrightarrow{E}_i^t$. Note that the all the calculations are performed in GF (256) and $rank([\overrightarrow{V}_{i,1}^t, \overrightarrow{V}_{i,2}^t, ..., \overrightarrow{V}_{i,K_i^t}^t]) = K_i^t$, i.e., all the vectors associated with the existing packets are linearly independent with each other. This follows from the following pair of observations: *1)* the original packets in each node are linearly independent from each other; and *2)* a packet received by a node n_j is stored at the node if and only if the received packet is also linearly independent with respect to $\mathcal{U}_j^t$. Such a received packet is said to be an *innovative packet*.

b) Packet Reception: As in [2], we use the indicator variable $s_i^t = 1$ to denote that node n_i transmits in round t and $s_i^t = 0$ otherwise. A node n_j successfully receives an innovative packet P_i^t from node n_i in round t iff:

- $s_i^t = 1$: n_i transmits in round t,
- $(n_i, n_j) \in \mathcal{L}_\mathcal{T}$: there exists a transmission link from n_i to n_j,
- $s_k^t = 0, \forall n_k \in \mathcal{N}|(n_k, n_j) \in (\mathcal{L}_\mathcal{T} \cup \mathcal{L}_\mathcal{I} \setminus \{(n_i, n_j)\})$: all nodes $n_k \neq n_i$, for which there exists a transmission or collision link to n_j, do not transmit in round t,
- $rank([\overrightarrow{V}_{j,1}^t, \overrightarrow{V}_{j,2}^t, ..., \overrightarrow{V}_{j,K_j^t}^t, \overrightarrow{E}_i^t]) = K_j^t + 1$: the encoding vector of the received packet is linearly independent with all the existing vectors in n_j.

The first three conditions guarantee that n_j receives P_i^t successfully, and the last condition ensures that P_i^t is innovative for n_j. If all these conditions are met then the n_j stores P_i^t, i.e., $\mathcal{U}_i^{t+1} = \mathcal{U}_i^t \cup \{P_i^t\}$.

c) Transmission Scheduling: We define $\mathcal{PS}^t$ to be the set containing the sets of packets stored at each node $n_i \in \mathcal{N}$ at round t, i.e., $\mathcal{PS}^t = \{\mathcal{U}_1^t, \mathcal{U}_2^t, ..., \mathcal{U}_N^t\}$. We further define $\overrightarrow{h}^t$ to be the transmission policy at round t. $\overrightarrow{h}^t$ is an N vector containing all transmission indicator variables, i.e., $\overrightarrow{h}^t = [s_1^t, s_2^t, ..., s_N^t]'$. The objective of NC-CPR is to find a series of transmission policies $H_Q = (\overrightarrow{h}^1, \overrightarrow{h}^2, ..., \overrightarrow{h}^Q)$ so that $\mathcal{PS}^0 \overset{t_1}{\Rightarrow} \mathcal{PS}^1 \overset{t_2}{\Rightarrow} ... \overset{t_Q}{\Rightarrow} \mathcal{PS}^Q$. $\mathcal{PS}^Q$ is the set where $K_i^Q = M$ for all $\mathcal{U}_i^Q, n_i \in \mathcal{N}$. In other words, each node will have M linearly independent packets at transmission round Q. Every node $n_i \in \mathcal{N}$ can, therefore, recover $\mathcal{M}$ by solving M linear equations using the matrix $[\overrightarrow{V}_{i,1}^Q, \overrightarrow{V}_{i,2}^Q, ..., \overrightarrow{V}_{i,M}^Q]$. The minimum latency NC-CPR problem is to find a series of transmission policies H_Q that minimize Q.

III. NC-CPR Protocols

A. NC-CPR with Optimal Scheduling

An optimal scheduling minimizes the total number of transmission rounds so that total repair time is minimized. However, looking for an optimal scheduling is an NP-Hard problem. This can be proven if we set the batch size to 1, i.e., $M = 1$. In this case, NC-CPR will be the same as the CPR problem where nodes that have the packet send it to others without the packet. In [10] we have proven that the CPR problem for directed graph with collision and interference links is NP-Hard. Therefore NC-CPR with $M = 1$ is NP-Hard, which also means generally the NC-CPR problem is NP-Hard. We have the conclusion shown below:

Theorem 1: Minimum latency (optimal) scheduling problem for the NC-CPR is NP-Hard.

Since the optimal scheduling is NP-Hard, there is no polynomial-time optimal algorithm of the size of the input instance for NC-CPR to solve the problem optimally unless P=NP to achieve optimal performance. We therefore develop a lower bound for the optimal scheduling, i.e., the minimum number of transmission rounds is at least $\max_i\{M - |\mathcal{I}_i|\}$. The lower bound is true because each node can receive maximum of 1 packet in each transmission round, so if it is missing K packets, it takes minimum of K rounds to recover. We use the lower bound later as a performance benchmark for our NC-CCPR algorithm.

B. Centralized NC-CPR

Since the optimal scheduling for NC-CPR is NP-Hard, we propose a heuristic-based NC-CCPR algorithm. We assume that all the nodes have the exact one-hop and two-hop neighbors information, i.e., number of neighbors and how many innovative packets do they have. The rationale behind NC-CCPR is that during each transmission round, we want more nodes to receive innovative packets. We come up with four heuristics:

- A node with more packets should have higher chance to transmit. Intuitively, the more packets a node has, the better this node can help its neighbors.
- The fewer packets that a node's neighbors have, the higher chance the node should transmit.
- The more neighbors a node has, the higher chance the node should transmit.
- The fewer two-hop neighbors a node has, the higher chance the node should transmit. This is due to the fact that we restrict simultaneous transmissions within two-hop neighbors. The fewer two-hop neighbors a node has, the more likely that more nodes can transmit in the same transmission round.

We let $\mathcal{T}^t$ be the transmit node set at round t and define $\overrightarrow{V}_{EP}$ as the vector containing the number of existing packets of each node in set $\mathcal{T}^t$. Similarly, $\overrightarrow{V}_{NP}$ is the vector containing the number of one-hop neighbors' packets of each node in set $\mathcal{T}^t$, $\overrightarrow{V}_N$ is the vector of the number of one-hop neighbors in each node, and $\overrightarrow{V}_{THN}$ is the vector of the number of two-hop

neighbors in each node respectively. Based on the heuristics, we define $Rank$ as

$$
\begin{aligned}
Rank \quad = \quad & Index_{\mathcal{T}^t}(\max\{C_1 * \overrightarrow{V}_{EP} - C_2 * \overrightarrow{V}_{NP} \\
& + \quad C_3 * \overrightarrow{V}_N - C_4 * \overrightarrow{V}_{THN}\}),
\end{aligned} \quad (1)
$$

in which $Rank$ returns the index of the highest ranked node in $\mathcal{T}^t$ and C_1, C_2, C_3 and C_4 are parameters to be tuned to optimize performance. The Algorithm below shows how we find the transmission policy $\overrightarrow{h}^t$.

Algorithm 1: NC-CCPR Transmission Policy

Input : $G = (\mathcal{N}, \mathcal{L}_{\mathcal{T}}, \mathcal{L}_{\mathcal{I}})$, $\overrightarrow{h}^{t-1}$
Output: $\overrightarrow{h}^t$
$\mathcal{T}^t \leftarrow \mathcal{N} \setminus \{\forall n_i \in \mathcal{N} | h^{t-1}(i) = 1\}$;
$\overrightarrow{h}^t \leftarrow \overrightarrow{\emptyset}_{N \times 1}$;
while $\mathcal{T}^t \neq \emptyset$ **do**
    $n_k \leftarrow \mathcal{T}^t(Rank(\mathcal{T}^t))$;
    $neighborSet \leftarrow \forall n_i \in \mathcal{T}^t | (n_i, n_k) \in \mathcal{L}_{\mathcal{T}}$;
    $\mathcal{T}^t \leftarrow \mathcal{T}^t \setminus (neighborSet \cup \{n_k\})$;
    $twoHopSet \leftarrow$
    $\forall n_i \in \mathcal{T}^t | \forall n_j \in neighborSet, (n_i, n_j) \in (\mathcal{L}_{\mathcal{T}} \cup L_I)$;
    $\mathcal{T}^t \leftarrow \mathcal{T}^t \setminus twoHopSet$;
    $\overrightarrow{h}^t(k) \leftarrow 1$;
end
$return \ \overrightarrow{h}^t$;

Note that we prevent highly ranked nodes from transmitting all the time by eliminating the nodes that transmitted in round $t-1$ from $\mathcal{T}^t$. We also note that the output transmission nodes from Algorithm 1 are collision-free, i.e., nodes that transmit in the same transmission round are out of the transmission or interference range from each other. This may lead to a possible performance gap between the protocol and the optimal performance. We leave the study of a centralized protocol that allows collision for future work. However, later in the NC-DCPR protocol, collision is allowed and the protocol addresses various practical concerns.

Through extensive simulation, we find that a wide range of values are available for the four parameters to generate satisfactory performance. We also note that C_1 and C_2 have the most weight on the result, which suggests that the first two heuristics are more important. Therefore we only consider the first two heuristic for the NC-DCPR protocol. In fact, in practical conditions, it is much easier to get one-hop information than multi-hop.

We compare the performance of NC-CCPR together with the lower bound in Fig. 2. We have 49 nodes uniformly distributed in a $1000 \times 1000 m^2$ square area. The transmission range is $250m$ while the interference range is $280m$. The MBMS has i.i.d. loss rate 0.3. It is possible that some nodes may be located out of the transmission range of the other nodes so that transmissions can not reach them. In this case, these nodes cannot be repaired using CPR. Therefore we only consider the case that no nodes are isolated, i.e., the nodes form a connected graph.

We also consider another scheme, *Random select* in Fig. 2. *Random select* uses similar method as in NC-CCPR to find
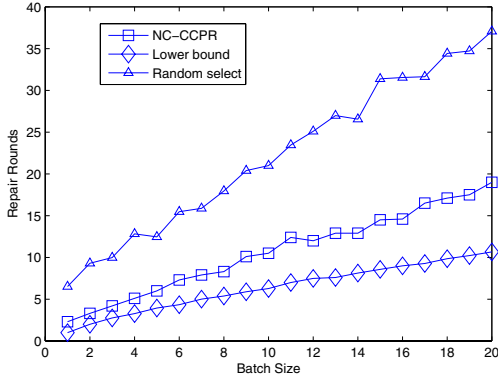
Fig. 2. Performance comparison of NC-CCPR with lower bound, and the *Random select* scheme. Node number is 49.

multiple transmitting nodes. However, nodes in this case are selected randomly without considering the $Rank$. It is obvious that NC-CCPR provides big improvement over random node selection. We can see that transmission rounds needed for the NC-CCPR algorithm is less than 2 times the lower bound. Therefore the ranking scheme is effective in finding proper candidate nodes for transmission.

C. Distributed NC-CPR

In practice, it is difficult for each node to get global information and it is also unclear how to synchronize the nodes efficiently. Therefore, we propose the NC-DCPR protocol based on the heuristics from NC-CCPR. It turns out that the proposed protocol provides great performance improvement over DCPR in [2] where NC is not used.

The batch based NC-DCPR implements NC in the same way as in NC-CCPR. When nodes have received a batch of packet from MBMS, they initiate the repairing process. The repairing goes from batch to batch. The header of the transmitted packet should contain the encoding vector, therefore the size of the header will be around M bytes, which makes the overhead of NC-DCPR $\frac{M}{B}$. We prefer smaller batch size M because reducing M leads to smaller overhead and shorter time it takes to wait for the starting of the repair process. The header also includes the number of innovative packets the transmitting node has. Using this information the receiving nodes can then estimate the total number of neighbors' packets, which is used for transmission scheduling as shown later.

Whenever a node is ready to send a packet, it waits for Transmit Wait Interval (TWI) time before transmission. TWI is made up of four parts, namely, Transmit Penalty (TP), Random Wait Interval (RWI), Self Oriented Wait Interval ($SOWI$) and Neighbor Oriented Wait Interval ($NOWI$). To be more specific, we have

$$TWI = TP + RWI + SOWI + NOWI, \qquad (2)$$

where $SOWI = \frac{C_1 \times M}{\#ExistingPackets}$ and $NOWI = \frac{C_2 \times (\#Neighbors'Packets)}{\#Neighbors \times M}$. C_1 and C_2 are two parameters to be tuned for optimal weights on the two heuristics. We illustrate

the functions of the four parts as follows. TP and RWI are used to reduce the chance that a highly ranked node transmits all the time. Whenever a node has just sent a packet, it waits for an additional TP time to send the next packet. $SOWI$ captures the heuristic that the more packets a node has, the less time it should wait to transmit. Similarly, $NOWI$ presents the heuristic that the fewer packets a node's neighbors have, the less time it should wait for the next transmission. Whenever a node receives a packet, it cancels its current scheduled transmission and wait for a new TWI time. In this way, transmission collision is further reduced. In the next section, we evaluate the performance of the NC-DCPR protocol through extensive simulation studies that validate the effectiveness of the protocol.

IV. EXPERIMENTAL EVALUATION

A. Simulation Setup

We consider a $1000 \times 1000 m^2$ square network on which nodes are either uniformly distributed or clustered depending on the network topology. 3G MBMS [1] sends at its maximum rate of $384kbps$. The repair deadline for the batch based NC-DCPR protocol is the time that nodes receive one batch of packets from MBMS, i.e., the maximum allowed repair time $epoch = \frac{8 \times B \times M}{384} ms$. We consider two MBMS packet loss models with average loss rate L: Identical Independent Distributed (IID) and Spatial & Temporal Locality (STL). For the IID model, nodes within the network have i.i.d. loss rate L. For the STL model, nodes within the $\frac{1000}{\sqrt{2}} \times \frac{1000}{\sqrt{2}} m^2$ square have average loss rate $0.75L$ while nodes outside of the square have average loss rate $1.25L$, which captures the spatial packet loss characteristic. We also use the Gilbert model [11] in STL to capture the burst packet loss characteristic. To show how topology affects the performance, we use a special biased loss model for the clustered topology, where in each of the cluster, some of the packets are lost by all of the nodes. This captures the scenario that some nodes further away from the MBMS base station may lose packets at the same time. The simulation is performed using QualNet in which we set all the nodes to work in 802.11g broadcast mode where ARQ is disabled.

The parameter TWI is tuned under different settings for better performance. The settings we explore are: $M = \{5, 10, 50, 100\}$, $L = \{0.1, 0.3, 0.5, 0.7\}$, $B = \{100, 500, 1000\}$, loss model={IID, STL}, and Topology={Homogeneous, Cluster}. By varying the parameters in (2), we find that many combinations give good result. We choose one of the best combinations [1, 2, 4, 0.5], for C_1, C_2, TP, and the maximum value of RWI respectively, which is used throughout the simulations. The repair time shown in the next subsection is the time from the starting of an epoch till all the nodes have received enough innovative packets to perform decoding. The NC decoding time is not included. However as shown in [12], the NC decoding time can be covered by the repair packets transmission time by using *progressive decoding* and therefore is ignored for the sake of clearly showing the performance of the protocol.
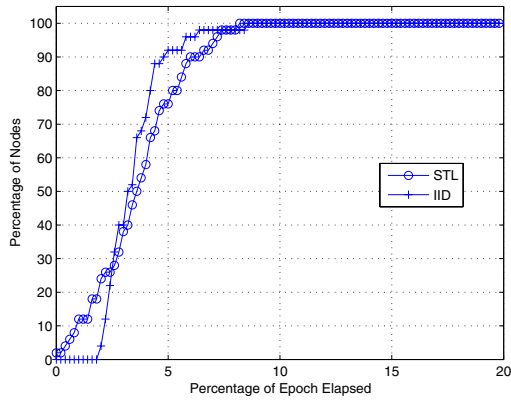
Fig. 3. CDF of NC-DCPR protocol repair time. Comparison between loss models STL and IID. MBMS loss rate is 0.3. Batch size is 100 and packet size is $1000bytes$.
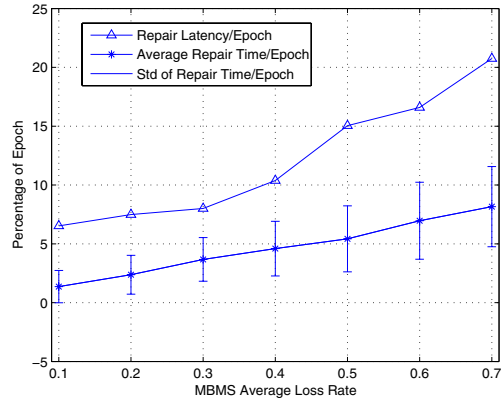

Fig. 4. Repair latency, average repair time and standard deviation of repair time under different MBMS average loss rate. Batch size is 100. Packet size is $1000bytes$.
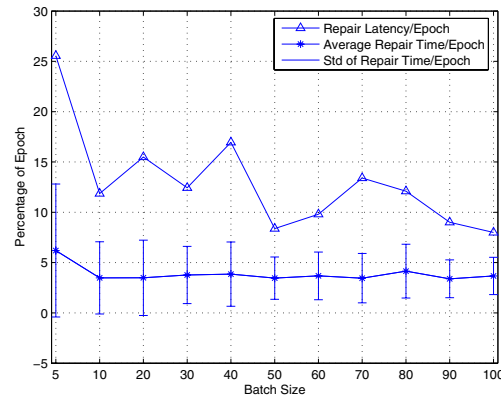

Fig. 5. Repair latency, average repair time and standard deviation of repair time under different batch size. MBMS loss rate is 0.3. Packet size is $1000bytes$.

B. Protocol Efficiency

a) Loss Model and Loss Rate: We uniformly put 50 nodes in the $1000 \times 1000m^2$ square network. We choose the repair batch size as 100, packet size $1000bytes$ and the MBMS average loss rate $L = 0.3$. In this case, *epoch* is around $2s$. Fig. 3 shows the Cumulative Distribution Function (CDF) of the repair time of all the nodes. It turns out that all the nodes are recovered within 10% of the *epoch*. We also conducted the simulation for CPR without NC with similar settings, which yielded a repair latency of around 40% of the *epoch*. Therefore in terms of repair time, NC-CPR outperforms CPR without NC significantly. Note that for the STL model more nodes finish repairing in very short time. This is due to the fact that loss distribution in STL model is not homogeneous. It is possible that some nodes lose very few packets while some lose much more. Therefore some nodes can be recovered very quickly while others may take longer time to repair. The average repair time for the IID and STL models are $72.07ms$ and $76.56ms$, which implies that the degree of homogeneity of packet loss influences the repair time, but in a limited manner. We use the more practical STL model for the following discussions.

By varying the MBMS packet loss rate from 0.1 to 0.7, Fig. 4 shows how loss rate affects the performance. The result is intuitive since higher loss rate incurs longer repair time. However, it is worthy to note that even when the loss rate is 0.7, all nodes can be recovered around 20% of the *epoch*, which further shows the effectiveness of the protocol.

b) Batch Size and Packet Size: Batch size and packet size are two important parameters for our protocol. They affect the time to wait before repairing and the repair latency. Again we put 50 nodes in the $1000 \times 1000m^2$ square network uniformly. With the loss rate 0.3 and loss model STL, Fig. 5 and Fig. 6 show how batch size and packet size affect the protocol performance. It turns out when batch size is larger than 10, it does not affect the protocol much. Since smaller batch size means shorter time to wait to start the repairing, we prefer small batch size for the protocol.

By analyzing Fig. 6, we observe that when packet size is 100, it takes relatively longer time to recover all the nodes while for other packet sizes, the repair time is shorter and similar. This is because while transmission time and *epoch* are reduced with smaller packet size, TWI is not reduced. Therefore comparing with the packet transmission time, TWI plays a more important role in the total repair time, making the relative repairing time longer than when packet size is large.

c) Network Topology and Density: We show how topology affects the performance of the protocol through Fig. 7. We consider three topologies: homogeneous, clustered and bias clustered. In the homogeneous topology, nodes are uniformly distributed in the $1000 \times 1000m^2$ square network while for the clustered topology, nodes are grouped into two clusters, one with 24 and the other with 25 nodes. There is one bridge node connecting the two clusters. Nodes in different clusters are out of the transmission range from each other. Communication between the two clusters can only be done through the bridge node. The MBMS loss model for the homogeneous and the
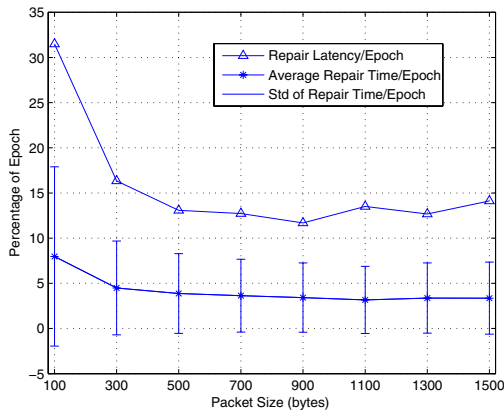
Fig. 6. Repair latency, average repair time and standard deviation of repair time under different packet size. MBMS loss rate is 0.3. Batch size is 10.
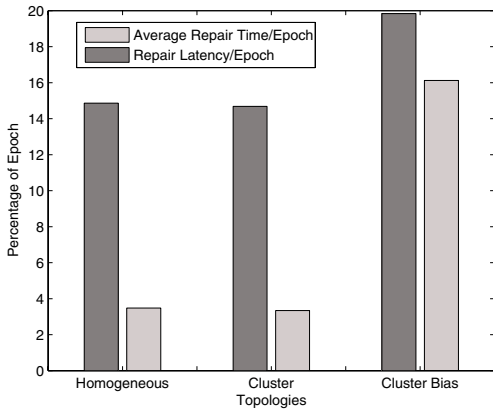

Fig. 7. Repair latency and average repair time under different topologies. MBMS loss rate is 0.3. Batch size is 10. Packet size is $1000bytes$.
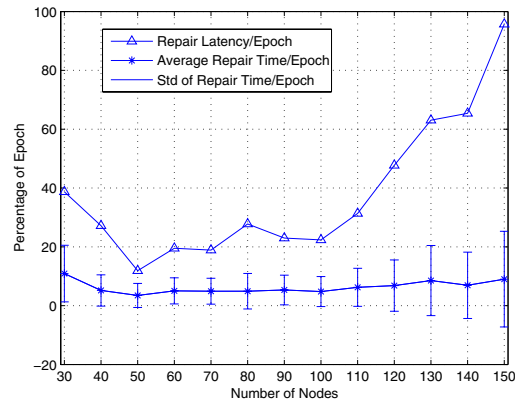

Fig. 8. Repair latency, average repair time and standard deviation of repair time under different number of nodes. MBMS loss rate is 0.3. Batch size is 10. Packet size is $1000bytes$.

clustered topologies is STL. We can see that the performance under the two topologies are similar. This is easy to understand because most likely nodes in each of the clusters contain all the original messages so that repairing can be done within the same cluster. However, when we change the loss model so that all the nodes within one cluster lose certain packets and only through the other cluster can the packets be recovered, repair time is increased. This is the case for the biased topology. It is clear that the performance deteriorates under this condition.

In Fig. 8, we can see that when the network density is larger than 100, repair time increases sharply because of the increase of collisions. When the density is too small, nodes are far too apart from each other and so the repair time is also increased. In fact when there are only 20 nodes in the network, not all the nodes can be recovered because some nodes are out of the transmission range of all the other nodes who have the needed packets, making the network a disconnected graph.

V. CONCLUSION

Based on the concept of CPR [2], we propose a NC-based approach, NC-CPR, to address the packet loss problem in MBMS. We show that the optimal transmission scheduling for NC-CPR is NP-Hard and provide a lower bound as performance benchmark. Based on the analysis of the protocol, we propose performance improving heuristics, which are implemented in the NC-CCPR and the NC-DCPR protocol. Extensive simulations show that the protocols are efficient in information dissemination. Realistic concerns are addressed in NC-DCPR to enhance its adoption for practical use.

REFERENCES

[1] *Technical Specification Group Services and System Aspects: Multimedia Broadcast/Multicast Service (MBMS) user services; stage 1*, *3GPP TS.26.246 version 6.3.0 Release 6* , March 2006.
[2] S. Raza, D. Li, C.-N. Chuah, and G. Cheung, " Cooperative Peer-to-Peer Repair for Wireless Multimedia Broadcast", *IEEE ICME 2007*, pp.1075-1078, July 2007.
[3] S. Raza, G. Cheung, and C.-N. Chuah, "DICoR: Distributed Interference-Aware Cooperative Repair of Multimedia Broadcast Losses", *submitted to IEEE SECON 2008, San Francisco, CA, June 2008*.
[4] P. Sharma and S.-J. Lee and J. Brassil and K. Shin", "Distributed Communication Paradigm for Wireless Community Networks", *IEEE ICC 2005*, vol. 3, pp. 1549-1555, 2005.
[5] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network Information Flow", *IEEE Trans. Info. Theory*, 2000.
[6] T. Ho, R. Koetter, M. Mdard, M. Effros, J. Shi, and D. Karger, "A Random Linear Network Coding Approach to Multicast", *IEEE Transactions on Infomation Theory*, vol. 52, pp. 4413 - 4430, Oct. 2006.
[7] C. Gkantsidis and P. R. Rodriguez, "Network Coding for Large Scale Content Distribution", *IEEE Infocom 2005*, vol. 4, pp.2235-2245, March 2005.
[8] S. Deb, M. Medard, C. Choute, "Algebraic Gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering", *IEEE Transactions Infomation Theory*, vol. 52, pp. 2486-2507, June 2006.
[9] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, " A Network Coding Approach to Energy Efficient Broadcasting: From Theory to Practice", *IEEE Infocom 2006*, pp.1-11, 2006.
[10] S. Raza, G. Cheung, and C-N. Chuah, "On the Complexity of Cooperative Peer-to-Peer Repair Problem with Collisions", *Technical Report ECE-CE-2007-2, Computer Engineering Research Laboratory, UCDavis, URL: http://www.ece.ucdavis.edu/cerl/techreports/2007-2/*, June 2007.
[11] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A Markov-Based Channel Model Algorithm for Wireless Networks", *Wireless Networks*, vol. 9, no. 3, pp. 189-199, 2003.
[12] M. Wang and B. Li, "Lava: A Reality Check of Network Coding in Peer-to-Peer Live Streaming", *IEEE Infocom 2007*, pp. 1082-1090, May, 2007.