

Structured Network Coding and Cooperative Local Peer-to-Peer Repair for MBMS Video Streaming

Xin Liu ^{#1}, Gene Cheung ^{*2}, Chen-Nee Chuah ^{#3}

[#] *Department of Electrical & Computer Engineering, University of California, Davis
Kemper Hall, Davis, CA 95616-5294*

¹xyzliu@ucdavis.edu ³chuah@ucdavis.edu

^{*} *Hewlett-Packard Laboratories Japan*

3-8-13, Takaido-higashi, Suginami-ku, Tokyo 168-0072, Japan

²gene-cs.cheung@hp.com

Abstract—By providing coding ability at intermediate nodes, network coding has been shown to improve throughput in wireless broadcast/multicast networks. Considering a scenario where wireless ad-hoc peers cooperatively relay packets to each other to recover packets lost during MBMS broadcast, we show that by first imposing coding structures globally and then selecting the appropriate types within the structures locally, network coding can be optimized for video streaming in a rate-distortion manner. Experimental results show that our proposed scheme can improve video quality noticeably, by up to 19.71dB over un-repaired video stream and by up to 8.34dB over video stream using traditional unstructured network coding.

I. INTRODUCTION

Due to wireless cellular networks' limited bandwidths and lossy transmission channels, high-quality video streaming over these networks is known to be difficult [1]. The advent of Multimedia Broadcast Multicast Service (MBMS) [2], in 3GPP specification version 6 or later, means media content can now be delivered to multiple users simultaneously and efficiently via a shared physical channel. However, it also means previously developed feedback-based loss recovery schemes like [1] for point-to-point streaming are no longer applicable due to the well-known NAK implosion problem, making video streaming over MBMS even more challenging.

To address the problem, we have previously proposed a *Cooperative Peer-to-Peer Repair* (CPR) framework [3] for a community of wireless peers with both cellular and 802.11 network interfaces. The idea is simple: having each correctly received a different subset of packets from MBMS broadcast (due to different channel conditions experienced), an ad-hoc network of peers can then locally broadcast their packets via 802.11 to cooperatively recover lost MBMS packets. Using our developed heuristics, we showed in [3] that significant packet recovery can be achieved under reasonable network settings. Moreover, if we permit each peer to perform *Network Coding* (NC) [4]—linearly combining payloads of received packets in $GF(O)$, where O is the field size—before forwarding packets, we showed in [5] that even further performance gain in packet recovery can be achieved.

Compared to its cellular counterpart, an 802.11 network interface requires much more power to establish and maintain connections [6], [7], and hence powering both interfaces continuously for the entire duration of a long video stream is not practical for a lightweight, battery-powered mobile device. For

the purpose of CPR packet recovery then, it is more sensible to instead activate the 802.11 interface for only duration τ in every period T , where τ and T together determine the fraction of the 802.11 bandwidth available for peer-to-peer packet transmissions. In this energy-limited scenario, the more challenging research problem is: given the fraction of 802.11 bandwidth available for peer-to-peer transmissions, how to distributedly optimize CPR at each peer so that the resulting video distortions at the peers are minimized?

In this paper, we present a novel rate-distortion optimized, NC-based, cooperative video stream repair strategy for the energy-limited scenario. Our strategy operates in two steps. First, the media source defines an NC structure globally, so that packets of more important frames can be recovered with appropriately higher probabilities than less important ones for the average peer. Second, at a peer's transmission opportunity, given its available partial state information at hand about its neighbors, a peer selects a type within the defined NC structure for NC-coded packet transmission, minimizing its neighbors' distortions.

Compared to our most recent work [8], where we assumed zero state information exchanged among peers and the NC structure optimization was performed at the media source for an average peer, in this paper, we assume the more likely scenario where some state information, piggybacked on top of NC-coded repair packets, are propagated among peers and each of the peers can utilize the state information to make smarter packet transmission decision. Experiments showed that our two-step optimized structured network coding scheme improved video quality by up to 19.71dB over unrepaired video stream, by up to 8.34dB over video stream using unstructured network coding scheme. In comparison to the scheme in [8], the use of partial state information brought us up to 1.16dB performance improvement.

The outline of the paper is as follows. In Section II, we overview related works. In Section III, we discuss our chosen source and network models. We differentiate unstructured and structured network coding, the latter of which is used in our optimization framework shown in Section IV. Based on these discussions, we present our optimization framework in Section V. We explain our experimentation and results in Section VI. We conclude in Section VII.

II. RELATED WORK

Due to the aforementioned NAK implosion problem, many video streaming strategies over MBMS [9] have forgone feedback-based error recovery schemes like [1] and opted instead for Forward Error Correction (FEC)-based schemes like Raptor Codes. While FEC can certainly help some MBMS receivers recover some packets, receivers experiencing transient channel failures due to fading, shadowing, and interference still suffer great losses. Nevertheless, media source should perform some optimization to reduce the loss impact. In this work, we assume content source will first perform *reference frame selection* [10] during H.264 [11] video encoding so that inter-frame dependencies are minimized subject to an encoding rate constraint.

Network coding (NC) [4] has been an active research topic, and recent works [12], [13], [14] have attempted to jointly optimize video streaming and NC. [13] discussed a rate-distortion optimized NC scheme on a packet-by-packet basis for a wireless router, assuming perfect state knowledge of its neighbors. Though the context of our CPR problem is different, our formulation can be viewed as a generalization in that our optimization is on the entire Group of Pictures (GOP), while [13] is performed greedily per packet.

[12] discussed a hierarchical NC scheme where a layered structure is applied to a scalable, layer-coded video stream. Our formulation is more general in that our source dependency model is a directed acyclic graph (DAG), while the model in [12] is the more restricted dependency chain.

[14] discusses the application of Markov Decision Process [15] to NC, in which NC optimization and scheduling are centralized at the access point or base station. Like [13] they require complete state information assuming reliable ACK/NAK schemes, which has yet been shown to be scalable to large number of peers. In our CPR work, we instead consider fully distributed peer-to-peer repair without assuming full knowledge of state information of peers, and optimize the solution using a pre-determined NC structure.

III. MODELS AND REFERENCE FRAME SELECTION

A. Source Model

We assume the media source first performs *reference frame selection* during encoding of H.264 video [10], such that the inter-dependencies of frames in a GOP are minimized. In brief, the optimization works as follows. We first assume that each GOP is composed of a starting I-frame followed by $M - 1$ P-frames. Each P-frame can choose among a set of previous frames for motion compensation (MC), where each choice results in a different encoding rate and different dependency structure. If we now assume that a frame is correctly decoded only if it is correctly received and the frame it references is correctly decoded, then this choice also leads to a different correctly decoded probability. Using P-frames' selections of reference frames, [10] sought to maximize the expected number of correctly decoded frames given an encoding rate constraint.

Note that though H.264 [11] specification is more general and permits each coding block in a P-frame to individually choose a matching block in one of a number of previous frames for MC, we restrict all blocks in a given P-frame to reference a single previous frame. [10] showed that the streaming benefit outweighed the cost in coding restriction.

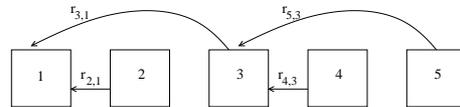


Fig. 1. DAG Source Model for H.264 Video with Reference Frame Selection

After the content source performed reference frame selection, we can now model M frames in a GOP, $\mathcal{F} = \{F_1, \dots, F_M\}$, as nodes in a *directed acyclic graph* (DAG) as shown in Figure 1, similarly done in [15]. Each frame F_i has an associated d_i , the resulting distortion reduction if F_i is correctly decoded. Each frame F_i points to the frame in the same GOP using which F_i performs MC. A frame F_i is correctly decoded iff F_i is correctly received, and all frames F_j 's preceding F_i , $j \prec i$, are correctly received. Frame F_i referencing frame F_j results in encoding rate $r_{i,j}$. We assume each frame is packetized into multiple RTP packets according to the frame size and Maximum Transport Unit (MTU) of the delivery network.

B. Network Model

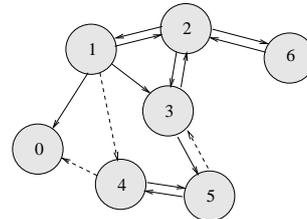


Fig. 2. Directed Graph Network Model: transmission and interference links are solid and dotted lines, respectively.

As done in [5], we assume that N peers in a wireless peer-to-peer network are modeled by nodes $1, \dots, N$ in node set \mathcal{N} in a directed graph $\mathcal{G} = (\mathcal{N}; \mathcal{L}_T, \mathcal{L}_I)$, and connectivities and interferences among nodes are modeled by links in link sets \mathcal{L}_T and \mathcal{L}_I , respectively. See Figure 2 for an example. A peer n_2 correctly receives a packet from transmitting peer n_1 iff: i) there exists a transmission link from n_1 to n_2 , i.e., $(n_1, n_2) \in \mathcal{L}_T$; and ii) no other nodes whose transmission or interference ranges include n_2 , i.e., $\forall n_i | (n_i, n_2) \in \mathcal{L}_T \cup \mathcal{L}_I$, is transmitting at the same time as n_1 . Notice that by this definition of successful transmission, we implicitly imply that the *broadcast mode* of 802.11 is used, where the transmission of a node can potentially be heard by all its neighbors.

Although the raw transmission rate of 802.11 is large, the peers need to contend for the shared medium for transmission in some distributed manner so that the occurrences of collision (simultaneously transmission of two nodes n_1, n_2 to a third

node n_3 where $(n_1, n_3), (n_2, n_3) \in \mathcal{L}_T$) and interference (simultaneously transmission of n_1, n_2 where $(n_1, n_3) \in \mathcal{L}_T$ and $(n_2, n_3) \in \mathcal{L}_I$) are reduced. Note that while transmission links \mathcal{L}_T are discovered through local message exchanges, interference links \mathcal{L}_I are unknown. To avoid collisions and interferences, we assume the transmissions follow a timer-based distributed protocol as done in [5]. Also, at the MAC layer of 802.11, when peer n senses a busy carrier, it backs off a random amount of time to further reduce collisions.

IV. NETWORK CODING FOR CPR

In this section, we describe how NC can be used in the CPR context. In particular, beyond the well known *Unstructured Network Coding*, we present *Structured Network Coding*, which can be partially decoded and can be optimized for video streaming in a rate-distortion manner.

A. Unstructured Network Coding

Suppose there are M original (native) frames $\mathcal{F} = \{F_1, \dots, F_M\}$ to be disseminated among N peers in a CPR setting. Each frame F_k is divided into multiple packets $\mathcal{P}_k = \{p_k^1, p_k^2, \dots, p_k^{B_k}\}$, where B_k is the number of packets frame F_k is divided into. We use \mathcal{P}^* to denote the set of all the packets in a GOP, i.e., $\mathcal{P}^* = \{\mathcal{P}_1, \dots, \mathcal{P}_M\}$. Hence there are $P = |\mathcal{P}^*| = \sum_{i=1}^M B_i$ packets to be disseminated. Using NC, each peer n can generate and transmit a NC packet q using a linear combination of its set of received MBMS native packets \mathcal{G}_n and its set of received NC packets \mathcal{Q}_n as follows:

$$q = \sum_{p_k^i \in \mathcal{G}_n} a_k^i p_k^i + \sum_{q_m \in \mathcal{Q}_n} b_m q_m \quad (1)$$

$$= \sum_{p_k^i \in \mathcal{P}^*} c_k^i p_k^i, \quad (2)$$

where a_k^i 's and b_m 's, random numbers in $GF(O)$, are coefficients for the received native packets and for the received NC packets, respectively. Because each received NC packet q_m is itself a linear combination of native and NC packets, we can rewrite q as a linear combination of native packets only with *native coefficients* c_k^i 's as shown in (2). For unstructured network coding (UNC), a_k^i 's and b_m 's are always non-zero, and a peer can reconstruct all P native packets when P "innovative" native or NC packets are received, meaning that all the frames can be recovered. By innovative, we mean that *native coefficient vector* $\mathbf{v} = [c_1^1, \dots, c_1^{B_1}, \dots, c_M^1, \dots, c_M^{B_M}]$ of a newly received NC packet cannot be a linear combination of native coefficient vectors from the set of previously received innovative native or NC packets.

The downside of UNC is that if a peer n receives fewer than P innovative native or NC packets, then the peer cannot recover *any* native packets using the received NC packets. If the probability of receiving at least P innovative native or NC packets for many peers is low, then this is not a desired result.

B. Structured Network Coding

To address the aforementioned issue, we instead use structured network coding (SNC). By imposing structure in the

coefficient vector, we seek to partially decode at a peer even when fewer than P innovative native or NC packets are received. We accomplish that by forcing some chosen coefficients a_k^i 's and b_m 's to be zeroes during NC packet generation, so that when a peer receives m innovative packets, $m < P$, it can decode m packets (m linear equations for m unknowns). Thus some of the frames can be recovered.

More precisely, given the DAG source model described in Section III-A, we first define a series of X *SNC frame groups*, $\Theta_1, \dots, \Theta_X$, where $\Theta_1 \subset \dots \subset \Theta_X = \mathcal{F}$. Corresponding to each SNC frame group Θ_x is a *SNC packet type* x . Let $g(j)$ be index of the smallest frame group that includes frame F_j :

$$g(j) = \arg \min_{x=1, \dots, X} |\Theta_x| \quad \text{s.t. } F_j \in \Theta_x \quad (3)$$

Native packets of frame F_j are of SNC packet type $g(j)$. SNC type of a NC packet q is identifiable in the packet header as $\Phi(q)$. A peer n can then encode a NC packet $q_n(x)$ of type x , given peer's set of received or decoded native packets \mathcal{G}_n and set of received NC packets \mathcal{Q}_n , as:

$$q_n(x) = \sum_{p_k^i \in \mathcal{G}_n} U(g(k) \leq x) a_k^i p_k^i + \sum_{q_m \in \mathcal{Q}_n} U(\Phi(q_m) \leq x) b_m q_m, \quad (4)$$

where $U(c)$ evaluates to 1 if clause c is true, and 0 otherwise. In words, peer n constructs NC packet of SNC type x by linearly combining received or decoded native packets of frames in Θ_x and received NC packets of SNC type $\leq x$.

A peer n can recover all $\sum_{F_i \in \Theta_x} B_i$ packets in frame group Θ_x once it has received $\sum_{F_i \in \Theta_x} B_i$ innovative packets of SNC types $\leq x$. We call this recovery process *NC-decoding*. In the following section, we show how the frame groups are defined at the media source using our optimization framework.

V. OPTIMIZATION FRAMEWORK

Our proposed SNC optimization strategy has two steps. First, the media source defines a NC structure to minimize distortion for the average user assuming zero state information exchange among peers. Second, at each transmission opportunity a peer selects a type within the defined NC structure to encode and transmit given its available partial state information of its neighbors. We discuss the two steps in the following.

A. SNC Definition at Media Source

We assume that a media source uses MBMS to deliver each GOP of M frames in time duration T , called an *epoch*. Repairs of the current GOP take place during the next epoch; 802.11 interface of each peer is activated from *sleep mode* to *idle mode* [7] for the first τ seconds of the next epoch T , during which peers can transmit and receive CPR packets for GOP of the previous epoch. The initial playback buffer delay for each peer is therefore two epochs.

The media source first optimizes the NC structure for the *average peer* n with average connectivity, assuming that on average a peer is expected to have received R_n packets from

neighbors. Using the DAG source model from Section III-A, the expected distortion at an average peer n can be written as:

$$\Delta_n = D - \sum_{i=1}^M d_i \prod_{j \leq i} \alpha_n(j) \quad (5)$$

where D is the initial distortion of the GOP if no frames are received, and $\alpha_n(j)$ is the recovery success probability of frame F_j at peer n . $\alpha_n(j)$ itself can be written as:

$$\alpha_n(j) = (1-l)^{B_j} + \left(1 - (1-l)^{B_j}\right) S_n(j) \quad (6)$$

where l is the MBMS packet loss rate, and $S_n(j)$ is the probability of frame F_j being recovered at peer n through CPR given F_j was not initially successfully delivered via MBMS.

Suppose we are given SNC groups $\Theta_1, \dots, \Theta_X$. Frame F_j can be recovered if $\sum_{F_i \in \Theta_{g(j)}} B_i$ innovative packets of SNC types $\leq g(j)$ are received, or if $\sum_{F_i \in \Theta_{g(j)+1}} B_i$ innovative packets of SNC types $\leq g(j)+1$ are received, etc. If a node n sends a NC packet of type x with probability $\beta_n(x)$, we can approximate $S_n(j)$ as:

$$S_n(j) = Q(n, g(j)) + \sum_{y=g(j)+1}^X Q(n, y) \prod_{z=g(j)+1}^y (1 - Q(n, z-1)) \quad (7)$$

where $Q(n, x)$ is the probability that node n can NC-decode SNC type x by receiving $\sum_{F_i \in \Theta_x} B_i$ innovative native or NC packets. We approximate $Q(n, x)$ as:

$$Q(n, x) \approx \sum_{k=l \sum_{F_i \in \Theta_x} B_i}^{R_n} \binom{R_n}{k} \left(\sum_{i=1}^x \beta_n(i) \right)^k \left(\sum_{i=x+1}^X \beta_n(i) \right)^{R_n-k} \quad (8)$$

where $l \sum_{F_i \in \Theta_x} B_i$ is the expected number of lost packets of type x due to MBMS broadcast and needed CPR repairs.

Assuming CPR has perfect collision avoidance, R_n , the average number of packets a peer can receive in an epoch time, can be approximated as:

$$R_n = \frac{\gamma\tau}{L/C_{\max}} \left(\frac{E_n^T}{E_n^T + 1} \right)^{E_n^I} \quad (9)$$

where γ is the fraction of bandwidth used for packet transmission after collision avoidance, which is estimated via experimentation. L is the average size of a CPR packet. C_{\max} is maximum rate of IEEE 802.11 interface used for CPR. Therefore $\frac{\gamma\tau}{L/C_{\max}}$ is the maximum number of packets node n can receive during an epoch time without considering interference. $E_n^T = |S : \{\forall n_i | (n_i, n) \in \mathcal{L}_T\}|$ and $E_n^I = |S : \{\forall n_i | (n_i, n) \in \mathcal{L}_I\}|$ are the expected numbers of neighboring and interference nodes of node n , respectively. Both of them are estimated via actual experimentation. We assume each interfering node has the same fraction of time for transmission as its neighboring nodes, thus $E_n^T / (E_n^T + 1)$ is the probability that an interfering node does not transmit at a given time and $\left(\frac{E_n^T}{E_n^T + 1} \right)^{E_n^I}$ is the non-interference transmission probability of node n .

With our formulation shown in equations (5)–(9), the SNC optimization process is therefore to find the number of frame

groups X , composition of frame groups Θ_x 's, and the packet transmission probabilities $\beta_n(x)$'s of frame groups so that the average distortion of the GOP is minimized:

$$\min_{X, \{\Theta_x\}, \{\beta_n(x)\}} \Delta_n \quad (10)$$

It is clear that the search space is too large for an exhaustive search approach to get the optimal network coding structure. We first notice that the search space can be reduced by considering the DAG structure described in Section III-A. A frame F_j that precedes frame F_i must surely be as important as frame F_i , since without it F_i cannot be correctly decoded. When we assign frames to SNC types then, we will assign preceding frames with a smaller or equal SNC type than succeeding frames given the DAG structure.

Based on the reduced search space, we then perform a fast local search method as follows. We first assign M SNC types to the M frames in *topological order* according to the DAG structure, so that a frame F_j preceding F_i will have a SNC type smaller than F_i . For this NC structure, we exhaustively search the best $\beta_n(x)$ resulting in the smallest distortion using (10). We then find the best “merging” of parent and child frames—assigning the same SNC type to the merged group—according to the DAG, and search for the best $\beta_n(x)$ for each of the group so that the objective is most reduced. We continue until no such beneficial merging operation can be found.

B. SNC Type Selection at Peer

From the NC structure defined through the optimization process shown in the previous section, we get the SNC types and packet transmission probabilities that minimize the average distortion of a GOP. The NC structure is defined using average state information, i.e., the media source targets at an average peer and makes no use of local neighborhood information. In fact, during the CPR repairing process, partial state information can be exchanged between neighbors. By partial we mean we only consider estimated one-hop neighbor information. Note that the information received from neighbors can be stale, given state information, piggybacked on NC packets, are transmitted intermittently and distributedly. The more practical method is to estimate the present state based on the neighbor information received previously. With the partial neighbor information, peers decide which type of packets to transmit instead of using the pre-determined packet transmission probability. Intuitively this could potentially improve the performance. For example, if a node's neighbors have recovered a certain frame group, then the node will not consider transmitting a NC packet of that particular type.

We assume that in each of the NC packets, the packet header includes two pieces of state information: the *native packet reception report* contains initially which packets are successfully delivered from MBMS and the *NC group status report* which contains the number of innovative packets that are received in each of the NC groups. There are totally P packets in one GOP and there are X SNC types, therefore $\frac{P}{8}$ bytes are needed for the first report and $\frac{X \log_2 P}{8}$ bytes are needed for the second state report. Compared to the size of

a NC packet, the additional state information will not incur much overhead. We will see some concrete results on this in Section VI.

Given the pre-determined SNC types from media source, peers can utilize the partial state information shown above to further optimize the packet transmission probabilities, i.e., node n finds the SNC type that results in the minimum total distortion among all its neighbors and transmits a packet of that type. More specifically, we optimize the following expression:

$$\min_{u=\{1,\dots,X\}} \sum_{m \in \{n\text{'s neighbors}\}} \Delta_m^u, \quad (11)$$

where u is the SNC type to be decided for packet transmission. Similar to (5), Δ_m^u is written as:

$$\Delta_m^u = D - \sum_{i=1}^M d_i \prod_{j \leq i} \alpha_m^u(j). \quad (12)$$

Since node n has the partial state information from neighbor m , we have

$$\alpha_m^u(j) = \begin{cases} 1, & \text{frame } j \text{ has been received;} \\ S_m^u(j), & \text{otherwise,} \end{cases} \quad (13)$$

Note that the first line in (13) has two meanings: either all the packets in frame j are successfully delivered through MBMS or they have been repaired through CPR. They are inferred from the native packet reception report and the NC group status report respectively. $S_m^u(j)$ can be written similarly as before except for the superscript representing the packet type to be decided:

$$S_m^u(j) = Q^u(m, g(j)) + \sum_{y=g(j)+1}^X Q^u(m, y) \prod_{z=g(j)+1}^y (1 - Q^u(m, z-1)) \quad (14)$$

Since peers now have neighbor information, $Q^u(m, x)$ is updated as

$$Q^u(m, x) \approx \sum_{k=L_m^{x,u}}^{U_m^u} \binom{U_m^u}{k} \left(\sum_{i=1}^x \beta_m(i) \right)^k \left(\sum_{i=x+1}^X \beta_m(i) \right)^{U_m^u - k}, \quad (15)$$

where

$$L_m^{x,u} = \begin{cases} C_m^x - \lfloor R_m \frac{t}{\tau} \sum_{i=1}^x \beta_m(i) \rfloor - 1, & u = x; \\ C_m^x - \lfloor R_m \frac{t}{\tau} \sum_{i=1}^x \beta_m(i) \rfloor, & u \neq x, \end{cases} \quad (16)$$

Here $L_m^{x,u}$ is the number of innovative packets of type $\leq x$ node m needs to recover frame group x . C_m^x is the actual number of innovative packets of type $\leq x$ neighbor m misses at the time when the state report is sent from m . t is the time elapsed from the last received state report up to present. $\lfloor R_m \frac{t}{\tau} \sum_{i=1}^x \beta_m(i) \rfloor$ represents the estimated number of packets of type $\leq x$ neighbor m could receive during time interval t . If the transmitted packet type u is the same as x , then we assume that with the transmitted packet from node n , neighbor m could reduce the number of missing packets of type x by 1. Similarly, U_m^u is the total number of packets

neighbor m could possibly receive during the rest of the repair time. It is written as:

$$U_m^u = \lfloor R_m (1 - \frac{t'}{\tau}) \rfloor - 1. \quad (17)$$

where t' is the time elapsed from the beginning of the repairing up to present. $\lfloor R_m (1 - \frac{t'}{\tau}) \rfloor$ is the number of packets neighbor m could receive in the remaining time. Since node n transmits a packet to its neighbor m , the total number of packets neighbor m could receive is reduced by 1.

Note that in equations (15) and (16), we assume conservatively that node m 's other neighbors do not perform local optimization using partial state information, but instead are transmitting using the pre-determined transmission probability. This is due to the fact that to predict the optimization results of node m 's other neighbors and what packets will be received by neighbor m during the rest of the repairing process, we need global state information, which is very difficult to achieve in a distributed scenario.

VI. EXPERIMENTATION

A. Experimental Setup

In this section we show through experiments the benefit of our two-step optimization scheme over the traditional UNC scheme under various CPR bandwidths, i.e., various τ/T ratios. The MBMS media source is assumed to transmit at rate 220kbps and the MBMS broadcast packet loss rate is constant at 0.1. Two test video sequences are used for simulations: 300-frame MPEG class A *news* and class B *foreman* sequences, which are captured at 30fps and sub-sampled in time by 2. The GOP size is 15 frames: one I-frame followed by 14 P-frames. Quantization parameters used for I-frames and P-frames are 30 and 25, respectively. The H.264 codec used is JM 12.4, downloadable from [16]. We perform reference frame selection in [10] with encoding target rate equals 220kbps, resulting in a DAG describing the inter-frame dependencies as discussed in Section III-A. We set MTU to be 1000bytes. Each of the frame is packetized into multiple packets according to the MTU size. With the encoding rate, typically there are 40 packets in one GOP. As shown in Section V-B, the overhead incurred from adding the partial state information is at most 30bytes (assuming X takes the largest value, P).

B. Experimental Results

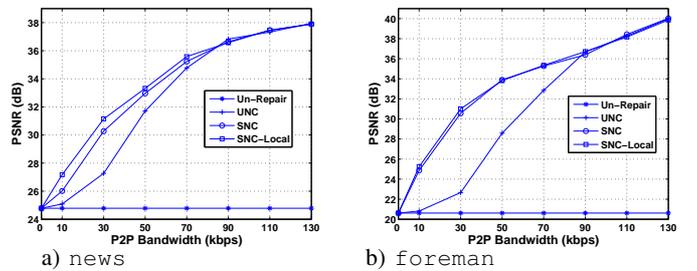


Fig. 3. PSNR for the *news* and *foreman* under various CPR transmission rate.

Fig. 3a and Fig. 3b showed the CPR bandwidth vs PSNR plot for the `news` and `foreman` sequences. The video qualities resulting from our proposed SNC scheme and UNC scheme were compared. We also compared the performance of the SNC scheme with and without type selection at peers. The un-repaired video quality was provided as a performance baseline. The CPR bandwidth varied from 0kbps up to 130kbps.

The video quality resulting from our proposed SNC scheme outperformed that of the traditional UNC scheme and the un-repaired video quality in almost all the transmission rates. Our SNC scheme provided 13.51dB PSNR improvement for the `news` sequence and 19.71dB PSNR improvement for the `foreman` sequence when the bandwidth was larger than 130kbps. For the UNC scheme, the peers needed possess $\sum_{j=1}^{15} B_j$ innovative native or NC packets before any repairing could be performed. However, for the SNC scheme, nodes could repair part of the frames as long as the received packets could help decode some SNC types of frames. This would be much less than the total number of packets. Therefore, when bandwidth was low, i.e., less than 90kbps, the performance of the SNC scheme was much better than the UNC scheme. For example at the transmission rate of 30kbps, the SNC scheme achieved 3.88dB gain over the UNC scheme for the `news` sequence and around 8.34dB gain for the `foreman` sequence. When the bandwidth was higher, i.e., larger than 90kbps, the number of received packets increased so that the UNC scheme recovered more packets and the performance of the two schemes became similar. Eventually both of the two schemes converged to the same best performance point when the bandwidth was larger than 130kbps, where both of them recovered all of the lost packets.

As for the comparison between the SNC schemes with and without packet type selection, the `news` sequence showed consistent observable performance improvement when the packet type selection was performed at the peers. We achieved a maximum of 1.16dB improvement at the CPR bandwidth of 10kbps. As for the `foreman` sequence, the improvement of doing type selection was small and we reaped a maximum gain of 0.44dB when the CPR bandwidth was 30kbps. It means for the `foreman` sequence, the packet transmission probabilities generated from the NC structure optimization at the media source was doing well in a probabilistic way. However, we note that with the partial state information, the SNC scheme was doing at least as good as the scheme without utilizing state information.

VII. CONCLUSIONS

In this paper, we present a novel rate-distortion optimized, NC-based, cooperative video stream repair strategy for 802.11 peer-to-peer networks. We focus on the case when the 802.11 network interfaces are only activated for a short amount of time periodically, and hence the repair bandwidth is low and a limited number of repair packets are transmitted. Specifically, reference frame selection is performed at the media source to minimize inter-frame dependencies in a GOP in H.264. We then perform a two-step NC structure optimization.

Firstly packets of video frames are mapped into a series of frame groups. After that, with partial state information, packet type selection is performed at each peer whenever it has the opportunity to transmit. In so doing, we show that our proposed scheme provides as large as 8.34dB video quality improvement over the UNC scheme when the CPR bandwidth is low, and up to 19.71dB improvement over the un-repaired video stream.

Although the discussion of the paper focused on the network scenario of 802.11 peer-to-peer repair of MBMS broadcast video, a carefully structured NC scheme is also useful for other combinations of peer-to-peer and broadcast technologies. For example, combination of bluetooth based peer-to-peer repair of MBMS broadcast video, or 802.11 peer-to-peer repair of DVB-H broadcast video.

REFERENCES

- [1] G. Cheung, W. t. Tan, and T. Yoshimura, "Double feedback streaming agent for real-time delivery of media over 3G wireless networks," in *IEEE Transactions on Multimedia*, vol. 6, no.2, April 2004, pp. 304–314.
- [2] *Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS) user services; stage 1 (Release 6) (3GPP TS.26.246 version 6.3.0*, March 2006.
- [3] S. Raza, D. Li, C.-N. Chuah, and G. Cheung, "Cooperative peer-to-peer repair for wireless multimedia broadcast," in *IEEE International Conference on Multimedia and Expo*, Beijing, China, July 2007.
- [4] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, "Network coding-based broadcast in mobile ad hoc networks," in *IEEE INFOCOM 2007*, Anchorage, AL, May 2007.
- [5] X. Liu, S. Raza, C.-N. Chuah, and G. Cheung, "Network coding based cooperative peer-to-peer repair in wireless ad-hoc networks," in *IEEE International Conference on Communications*, Beijing, China, May 2008.
- [6] A. Rahmati and L. Zhong, "Context-for-wireless: Context-sensitive energy-efficient wireless data transfer," in *ACM MobiSys*, San Juan, Puerto Rico, June 2007, pp. 165–178.
- [7] S. Chandra and A. Vahdat, "Application-specific network management for energy-aware streaming of popular multimedia formats," in *USENIX Annual Conference*, Monterey, CA, June 2002, pp. 329–342.
- [8] X. Liu, G. Cheung, and C.-N. Chuah, "Rate-distortion optimized network coding for cooperative video stream repair in wireless peer-to-peer networks," in *1st IEEE Workshop on Mobile Video Delivery*, Newport Beach, CA, June 2008.
- [9] J. Zfzal, T. Stockhammer, T. Gasiba, and W. Xu, "Video streaming over mbms: A system design approach," in *Journal of Multimedia*, vol. 1, no.5, August 2006, pp. 25–35.
- [10] G. Cheung, W.-T. Tan, and C. Chan, "Reference frame optimization for multiple-path video streaming with complexity scaling," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.6, June 2007, pp. 649–662.
- [11] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.7, July 2003, pp. 560–576.
- [12] K. Nguyen, T. Nguyen, and S.-C. Cheung, "Peer-to-peer streaming with hierarchical network coding," in *IEEE International Conference on Multimedia and Expo*, Beijing, China, July 2007.
- [13] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," in *IEEE 16th International Packet Video Workshop*, Lausanne, Switzerland, November 2007.
- [14] D. Nguyen, T. Nguyen, and X. Yang, "Multimedia wireless transmission with network coding," in *IEEE 16th International Packet Video Workshop*, Lausanne, Switzerland, November 2007, pp. 326–335.
- [15] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," in *IEEE Transactions on Multimedia*, vol. 8, no.2, April 2006, pp. 390–404.
- [16] "The TML project web-page and archive," <http://kbc.cs.tu-berlin.de/~stewe/vcegl/>.