Structured Network Coding and Cooperative Wireless Ad-Hoc Peer-to-Peer Repair for WWAN Video Broadcast

Xin Liu, *Student Member, IEEE*, Gene Cheung, *Senior Member, IEEE*, and Chen-Nee Chuah, *Senior Member, IEEE*

***Abstract*—In a scenario where each peer of an ad-hoc wireless local area network (WLAN) receives one of many available video streams from a wireless wide area network (WWAN), we propose a network-coding-based cooperative repair framework for the ad-hoc peer group to improve broadcast video quality during channel losses. Specifically, we first impose network coding structures globally, and then select the appropriate video streams and network coding types within the structures locally, so that repair can be optimized for broadcast video in a rate-distortion manner. Innovative probability—the likelihood that a repair packet is useful in data recovery to a receiving peer—is analyzed in this setting for accurate optimization of the network codes. Our simulation results show that by using our framework, video quality can be improved by up to 19.71 dB over un-repaired video stream and by up to 5.39 dB over video stream using traditional unstructured network coding.**

***Index Terms*—Cooperative peer-to-peer repair, network coding, wireless wide area network (WWAN) video broadcast.**

I. Introduction

With consumers' increasing demand for rich media contents and the ubiquity of mobile wireless access, deployments of various wireless multimedia services are fast emerging. To scale these services to large user bases, different wireless wide area network (WWAN) multimedia broadcast/multicast technologies have been proposed. For example, Multimedia Broadcast/Multicast Service (MBMS) [1] was introduced in UMTS cellular networks of 3GPP release 6.0 and later, which provides efficient point-to-multipoint multimedia delivery via a common cellular channel.

While the broadcast nature of the aforementioned WWAN multimedia distribution technologies enables scalable and bandwidth-efficient media delivery to a larger number of users via a common physical channel, it also has its share of technical challenges. First, previously developed feedback-based loss recovery schemes like [2] for point-to-point unicast streaming become infeasible in the broadcast scenario due to either the lack of a feedback channel, or the well-known NAK implosion problem [3] even if such feedback channel is available. Second, because broadcast systems are often optimized for the average channel [4] to maximize utility for the average user, packet losses are inevitable for the temporarily-worse-than-average users due to the unpredictable and time-varying nature of wireless channels, resulting in deteriorated video quality.

Given the recent popularity of multi-homed mobile devices [5]—devices with both 3G cellular and IEEE 802.11 wireless interfaces—one potential solution to the broadcast packet loss problem is for a group of interconnected peers listening to the same video stream to use their 802.11 interfaces to cooperatively perform *out-of-band* repair of 3G broadcast losses. This is the premise behind our previously proposed *cooperative peer-to-peer repair* (CPR) framework [6] to combat WWAN packet losses. Having each correctly received a different subset of packets from WWAN 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 WWAN packets. Using our developed heuristics, we showed in [6] that significant packet recovery can be achieved. Moreover, if we permit each peer to perform *network coding* (NC) [7]—linearly combining payloads of received packets in Galois Field $GF(O)$ where $O = 2^q$ is the field size and q is a positive integer—before forwarding packets, we showed in [8] that even further performance gain can be achieved.

Compared to its cellular counterpart, an 802.11 interface requires much more power to establish and maintain connections [9]–[11], and as a result, having both 3G and 802.11 interfaces activated constantly may not be feasible for lightweight battery-powered handheld devices consuming lengthy videos. To address the power consumption issue, we have previously imposed structures on NC [12], [13] to optimize repaired video quality given an energy budget.

In our previous works, we assumed that all peers in the same ad-hoc network are watching the same video; i.e., all available 802.11 bandwidth can be used to repair a single video stream. In practice, however, different users are likely watching different streams, and as a result, multiple streams (multi-stream) need CPR to improve broadcast video simultaneously. Fig. 1 illustrates the multi-stream scenario where different peers are watching different streams a, b, and c. Since each peer now needs to relay CPR packets of streams they are not watching, the network resource allocated to each stream is reduced. In this paper, we address this more realistic and more challenging scenario.

Manuscript received August 25, 2008; revised January 08, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. James E. Fowler.

X. Liu and C.-N. Chuah are with the Department of Electrical and Computer Engineering, University of California, Davis, Davis, CA 95616-5294 USA (e-mail: xyzliu@ucdavis.edu; chuah@ucdavis.edu).

G. Cheung is with Hewlett-Packard Laboratories Japan, Tokyo 168-0072, Japan (e-mail: gene-cs.cheung@hp.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2009.2017636

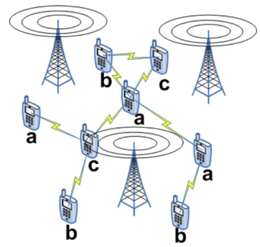

Fig. 1. Illustration of multi-stream scenario cooperative peer-to-peer repair.

Specifically, we present a rate-distortion optimized, NC-based, CPR solution for the multi-stream scenario to improve WWAN broadcast video quality. Our contributions are the following.

1) We propose a two-step NC optimization framework: 1) *global NC structure optimization*, where the media source defines an optimal NC structure globally based on the source's estimated average peer's network state, so that packets of more important frames can be recovered with appropriately higher probabilities for the average peer; 2) *local peer optimization*, where at a peer's transmission opportunity, given its available local state information at hand about its neighbors, a peer selects a stream and a NC type for packet transmission to minimize distortions particularly for its neighbors.
2) To facilitate accurate NC optimization, we estimate the *innovative probability* —likelihood that a received packet at a peer is useful for data recovery—in a computation-efficient manner.
3) We provided detailed simulations to verify our results, showing that our solution improves video quality significantly: by up to 19.71 dB over un-repaired video stream and by up to 5.39 dB over video stream using traditional unstructured NC schemes.

The outline of the paper is as follows. In Section II, we discuss the multi-stream system and our chosen source and network models. In Section III, we formally define unstructured NC and our proposed structured NC. In Section IV, we analyze packet innovativeness of receiving CPR packets at a given peer. Based on these discussions, we present our NC optimization framework in Section V. We explain our results in Section VI. We overview related works in Section VII and conclude in Section VIII, respectively.

II. System Architecture and Models

We first outline the architecture of our proposed broadcast video repair system. We then introduce two theoretical models used in our system optimization: 1) a video source model we use to optimize network coding for packet recovery, and 2) a network model used to schedule peer-to-peer packet repairs.

A. CPR System Architecture

We consider the scenario where N peers are watching broadcast video streams using their wireless mobile devices through the WWAN. The mobile devices are also equipped with *wireless local area network* (WLAN) interfaces, and the peers are physically located in close enough proximity that a peer-to-peer wireless ad-hoc network can be formed. The video streams can be live or stored content that are broadcasted from the *media source*; for simplicity, we denote media source to mean both a *media encoder* (where the video streams are encoded), and the actual video broadcasting entity over WWAN.

We first assume that the media source provides a total of S_{all} video streams. S_{all} varies due to different technologies, broadcast bandwidths, and operational constraints of the mobile video providers. Although S_{all} streams are available, not all streams will have audiences in a given ad-hoc network at a given time. Without loss of generality, we denote $\mathcal{S}^* = \{s^1, s^2, \ldots, s^S\}$ as the subset of S_{all} streams that have audience and $S = |\mathcal{S}^*|$. We assume that the media source can estimate the size S of the subset (rather than the actual subset $\mathcal{S}^*$ itself) based on its past history and inform the peers of its estimate using predefined fields in data packet headers.

Each peer n in the network watches one stream $\mathcal{S}(n) \in \mathcal{S}^*$ from the media source, and conversely each stream $s \in \mathcal{S}^*$ has a group of receiving peers $\mathcal{U}_s$. Peers in $\mathcal{U}_s$, each receiving a different subset of packets of stream s, can relay packets to others using WLAN interfaces to repair lost packets. This repair process is called CPR.

We assume that each peer is willing to relay repair packets of other streams; in return other peers will relay repair packets for the peer. We denote $\mathcal{A}_n$ as the set of streams of which peer n has received packets: either original video packets from the media source or CPR packets from peers, i.e., streams that peer n can repair via CPR. We use flags in CPR packet header to identify the stream a packet repairs. Whenever peer n has a transmission opportunity—a moment in time when peer n is permitted by a scheduling protocol (to be discussed) to locally broadcast a packet via WLAN, peer n selects one stream from $\mathcal{A}_n$ to construct and transmit a CPR packet.

B. Source Model

We use H.264 [14] codec for video source encoding because of its excellent rate-distortion performance. For improved error resilience, we assume the media source first performs *reference frame selection*[15] for each *group of picture* (GOP) in each stream separately during H.264 encoding. In brief, [15] assumes each GOP is composed of a starting I-frame followed by P-frames. Each P-frame can choose among a set of previous frames for *motion compensation*, where each choice results in a different encoding rate and different dependency structure. If we then assume that a frame is correctly decoded only if it is correctly received and the frame it referenced is correctly decoded, then this choice leads to a different correctly decoded probability. Using P-frames' selection of reference frames, [15] sought to maximize the expected number of correctly decoded frames given an encoding rate constraint.

After the media source performs reference frame selection for each GOP of each stream, we can model M^k frames in a GOP of a stream s^k, $\mathcal{F}^k = \{F_1^k, \ldots, F_{M^k}^k\}$, as nodes in a *directed acyclic graph* (DAG) as shown in Fig. 2, similarly done in [16]. Each frame F_i^k has an associated d_i^k, the resulting distortion reduction if F_i^k is correctly decoded. Each frame F_i^k points to the frame in the same GOP that it uses for motion compensation. Frame F_i^k referencing frame F_j^k results in encoding rate $r_{i,j}^k$.

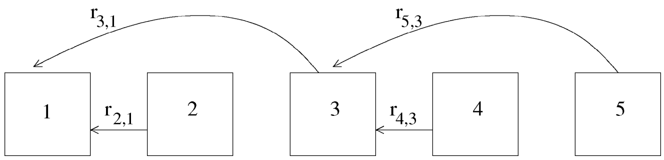

Fig. 2. Example of DAG source model for H.264/AVC video with reference frame selection.

We assume each frame F_j^k is packetized into *real-time transport protocol* (RTP) packets according to the frame size and *maximum transport unit* (MTU) of the delivery network. A frame F_j^k is correctly received only if all packets within F_j^k are correctly received.

We assume that the media source delivers each GOP of M^k frames of stream s^k in time duration Y^k. Y^k is also the *repair epoch* for s^k, which is the duration in which CPR completes its repair on the previous GOP; i.e., peers exchange CPR packets for previous GOP of stream s^k during the current epoch. The playback buffer delay for peer n is hence two epochs. Given that our later discussion focuses on one stream s^k, for simplicity we drop the superscript and refer to frame F_i^k simply as F_i, etc.

C. Network Model

As done in [8], we assume the multi-homed devices of N ad-hoc peers watching WWAN video perform CPR in 802.11 broadcast mode, so that a transmitted WLAN packet can potentially be received by more than one neighbor. Note that though raw WLAN transmission rate like 802.11 is relatively large, peers need to contend for the shared medium for transmission in a distributed manner so that the occurrences of collision and interference are reduced. For brevity, we omit the discussion on a distributed algorithm [8] that schedules WLAN ad-hoc peer transmissions. We simply assume that the average peer n can receive R_n total repair packets successfully via CPR in one repair epoch, which varies depending on available WLAN resources for CPR (constrained by factors such as power [12], [13] and contending cross traffic).

III. Network Coding Based CPR

In this section, we first describe *unstructured network coding* (UNC), common in the literature, in the context of CPR. We then present *structured network coding* (SNC), a new technique where by imposing structures on NC, one can further optimize NC specifically for video streaming in a rate-distortion manner.

A. Unstructured Network Coding

We denote the traditional random NC scheme [17] as UNC, as compared to our proposed SNC. First, suppose peer n has a transmission opportunity and n selects stream s from $\mathcal{A}_n$ for transmission. Suppose there are M original (native) frames $\mathcal{F} = \{F_1, \ldots, F_M\}$ in a GOP of stream s to be repaired among peers in $\mathcal{U}_s$. Each frame F_i is divided into multiple packets $\mathcal{P}_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,B_i}\}$ of size W bits each. Here B_i is the number of packets frame F_i is divided into. Note that a peer adds padding bits to each packet so that each has constant size W bits; this is performed for NC purposes, similarly done in [18]. We denote $\mathcal{P}^*$ as the set of all packets in a GOP, i.e., $\mathcal{P}^* = \{\mathcal{P}_1, \ldots, \mathcal{P}_M\}$. There are a total of $P = |\mathcal{P}^*| = \sum_{i=1}^{M} B_i$ packets to be disseminated among peers in $\mathcal{U}_s$.

We denote $\mathcal{G}_n$ as the set of *native packets* of stream $\mathcal{S}(n)$ peer n received from media source. Denote $\mathcal{Q}_n$ as the set of *NC packets* of stream s peer n received from other peers through CPR. If the stream selected for transmission is the same as the stream peer n currently watches, i.e., $s = \mathcal{S}(n)$, then the NC packet q_n generated by peer n is represented as

$$q_n = \sum_{p_{i,j} \in \mathcal{G}_n} a_{i,j} p_{i,j} + \sum_{q_m \in \mathcal{Q}_n} b_m q_m = \sum_{p_{i,j} \in \mathcal{P}^*} c_{i,j} p_{i,j} \quad (1)$$

where $a_{i,j}$'s and b_m's, random numbers in $GF(O)$, are coefficients for the original packets and the received encoded NC packets, respectively. Because each received NC packet q_m is itself a linear combination of native and NC packets, we can rewrite q_n as a linear combination of native packets with *native coefficients* $c_{i,j}$'s as shown in (1).

If the stream selected for transmission $s \neq \mathcal{S}(n)$, then the NC packet is simply a linear combination of all NC packets of stream s received through CPR from other peers so far as follows:

$$q_n = \sum_{q_m \in \mathcal{Q}_n} b_m q_m = \sum_{p_{i,j} \in \mathcal{P}^*} c_{i,j} p_{i,j}. \quad (2)$$

For UNC, *all* packets of stream s, both native packets (if any) and received NC packets, are used for NC encoding, and a peer in $\mathcal{U}_s$ can reconstruct all P native packets of stream s when P innovative native or NC packets of stream s are received, and hence all frames can be recovered. By innovative, we mean that native coefficient vector $\mathbf{v} = [c_{1,1}, \ldots, c_{1,B_1}, \ldots, c_{M,1}, \ldots, c_{M,B_M}]$ of a newly received packet is not a linear combination of native coefficient vectors from the set of previously received innovative packets. When a peer has accumulated P innovative packets, it recovers all P native packets in the GOP by solving P linear equations, each equation corresponding to an innovative packet, itself a sum of native packets as shown in (1).

The downside of UNC is that if a peer n receives fewer than P innovative packets, this 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. This is indeed the case for multi-stream, where the CPR bandwidth is shared by all streams, as we will see in Section VI. Hence there is a need to derive an alternative NC strategy for multi-stream.

B. Structured Network Coding

To address the aforementioned issue, we propose to use 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 of stream s are received. We accomplish that by forcing some chosen coefficients $a_{i,j}$'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) so that a subset of video frames in a GOP can be recovered.

More precisely, given the DAG source model described in Section II-B, for stream s, we first define a series of X *SNC frame groups*, $\Theta_1, \ldots, \Theta_X$, where $\Theta_1 \subset \cdots \subset \Theta_X = \mathcal{F}$. 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 as follows:

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

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 $\Phi(q)$. Similar to UNC, when the stream selected for transmission is the same as the stream that peer n watches, i.e., $s = \mathcal{S}(n)$, then the 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 $\mathcal{Q}_n$ is written as

$$\begin{aligned} q_n(x) = & \sum_{p_{i,j} \in \mathcal{G}_n} U(g(j) \leq x)\, a_{i,j} p_{i,j} \\ & + \sum_{q_m \in \mathcal{Q}_n} U(\Phi(q_m) \leq x)\, b_m q_m = \sum_{p_{i,j} \in \Theta_x} c_{i,j} p_{i,j} \end{aligned} \tag{4}$$

where $U(\mathsf{c})$ 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 $\leq x$. Note that the encoded packet of frame group Θ_X, i.e., $q_n(X)$, in SNC is the same as q_n in UNC. Similarly, if the stream selected for transmission is different from the stream that peer n watches, i.e., $s \neq \mathcal{S}(n)$, the generated NC packet is

$$q_n(x) = \sum_{q_m \in \mathcal{Q}_n} U(\Phi(q_m) \leq x)\, b_m q_m = \sum_{p_{i,j} \in \Theta_x} c_{i,j} p_{i,j}. \tag{5}$$

A peer n_i can recover all $\sum_{F_i \in \Theta_x} B_i$ packets in frame group Θ_x of stream s once it has received $\sum_{F_i \in \Theta_x} B_i$ innovative packets of SNC types $\leq x$. Fig. 3 shows a possible frame group assignment for a GOP of 15 frames with three frame groups. The probability of decoding F_1 is much higher than the other frames in frame groups 2 and 3. Since generally first I-frame F_1 of a GOP is the most important, by recovering only F_1, a large distortion can already be reduced.

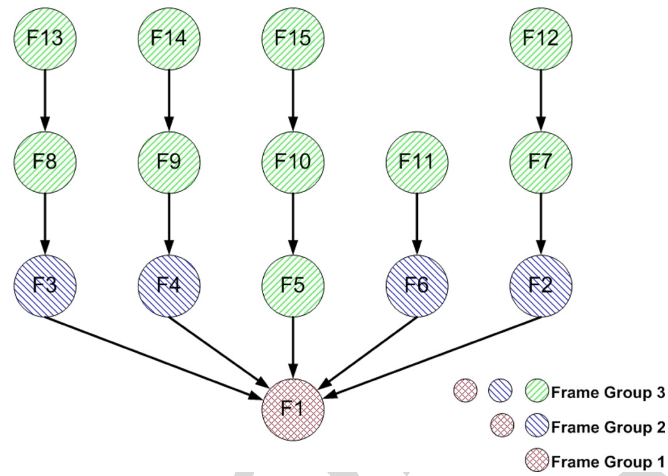

Fig. 3. DAG example with three frame groups.

IV. Packet Innovativeness

In this section, we estimate the innovative probability in a computation-efficient way. We first show a lower bound for the innovative probability for single stream case. Then by observing the differences between single and multi-stream, we estimate the innovative probability for the multi-stream case.

A. Innovative Probability for Single Stream

The exact computation of the NC packet innovative probability involves careful tracking of states of all peers in the CPR network. For example, [19] provided a complex innovative probability analysis for a gossip-based protocol, in which each peer in the network randomly selects another peer to send or to receive packets. Our CPR scenario is even more difficult in that each peer's transmission has multiple potential receivers because local WLAN broadcast is used. So instead of looking for an exact solution, we provide a simple and effective way of estimating the probability.

Suppose n_i transmits an NC packet to n_j using UNC. We denote B as the total number of packets needed to be disseminated for packet recovery; in the case of UNC, $B = P$. We also call B the *batch size*. We denote sets $\mathcal{S}_i = \{\mathbf{v}_i^1, \mathbf{v}_i^2, \ldots, \mathbf{v}_i^{k_i}\}$ and $\mathcal{S}_j = \{\mathbf{v}_j^1, \mathbf{v}_j^2, \ldots, \mathbf{v}_j^{k_j}\}$ as the native coefficient vectors of innovative native or NC packets in n_i and n_j before the transmission, respectively. Denote $\mathcal{SP}_i$ and $\mathcal{SP}_j$ as the subspaces spanned by the vectors in $\mathcal{S}_i$ and $\mathcal{S}_j$, respectively. Since vectors in $\mathcal{S}_i$ ($\mathcal{S}_j$) are linearly independent, they form a basis for subspace $\mathcal{SP}_i$ ($\mathcal{SP}_j$) with k_i (k_j) being the dimension of the subspace. n_j receiving an innovative packet means the coefficient vector associated with the received packet, together with vectors in $\mathcal{S}_j$, remain linearly independent. That means the innovative probability is also the probability that the dimension of $\mathcal{SP}_j$ increases.

We assume that the components in all native coefficient vectors take on values randomly chosen from $GF(O)$. This assumption is reasonable when peers are watching the same stream because in the UNC scheme all of the packets are treated equally and the encoding coefficients are also randomly chosen from $GF(O)$. We note that the assumption is less accurate at the beginning of the repairing process when the peers only have the chance to mix packets with neighbors close by. However it becomes more and more accurate with increasingly more packet mixing with peers. Let us define $P_{inv}^{k_i,k_j,B}$ as the *instantaneous innovative probability* of the received packet at peer n_j. We can summarize the lower bound for $P_{inv}^{k_i,k_j,B}$ with the following theorem.

Theorem 1: Assuming the dimensions of the subspaces spanned by the native coefficient vectors in peers n_i and n_j are k_i and k_j, then the instantaneous innovative probability of the NC packet transmitted from n_i to n_j has a lower bound as follows:

$$P_{inv}^{k_i,k_j,B} \geq \begin{cases} 1-\frac{1}{O}, & k_i > k_j \\ \left(1-\frac{1}{O}\right)(1-Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\}), & k_i \leq k_j \end{cases} \tag{6}$$

where $Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\}$ is the probability that the subspace spanned by vectors in $\mathcal{S}_i$ is a subset of the subspace spanned by the vectors in $\mathcal{S}_j$, which can be calculated as

$$Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\} = \frac{\prod_{t=0}^{k_i-1}(O^{k_j}-O^t)}{\prod_{t=0}^{k_i-1}(O^{B}-O^t)}. \tag{7}$$

Proof: We leverage [19, Lemma 2.1], which stated if the subspace spanned by native coefficient vectors in the transmitting peer is not a subset of the subspace spanned by the native coefficient vectors in the receiving peer, then the probability that the subspace dimension increases at the receiving peer, i.e., the innovative probability, is at least $1-1/O$. If dimension k_i of $\mathcal{SP}_i$ is larger than dimension k_j of $\mathcal{SP}_j$, then obviously $\mathcal{SP}_i \nsubseteq \mathcal{SP}_j$, and the first line of (6) follows.

The second line of (6) follows similar argument, and the key is to find $Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\}$ when $k_i \leq k_j$. Since $\mathcal{S}_i$ is a set of basis vectors for $\mathcal{SP}_i$, $Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\}$ is the same as the probability that each basis vector in $\mathcal{S}_i$ is also in $\mathcal{SP}_j$, i.e., $Pr\{\mathbf{v}_i^k \in \mathcal{SP}_j, \forall \mathbf{v}_i^k \in \mathcal{S}_i\}$. Since there are a total of O^B vectors over $GF(O)$, the first vector selected from $\mathcal{S}_i$ has O^B-1 possible choices excluding the zero vector. With k_j linearly independent vectors, there are O^{k_j} different vectors in subspace $\mathcal{SP}_j$. Then the probability that the first vector in $\mathcal{S}_i$ is in $\mathcal{SP}_j$ is $(O^{k_j}-1)/(O^B-1)$ where the "-1" in the numerator and denominator accounts for the zero vector. Similarly, the probability that the second vector in $\mathcal{S}_i$ is also in $\mathcal{SP}_j$ is $(O^{k_j}-O)/(O^B-O)$ where the "$-O$"accounts for vectors that are linear combinations of the first vector. Continue calculating the probabilities for the rest of the vectors in $\mathcal{S}_i$ and multiply all of them, we get the result for the second case. Combining the two cases, we have (6). □

Since our derivations are exact and the bound provided in [19, Lemma 2.1] is achievable, the result in Theorem 1 is tight and is achievable. Equation (6) shows the innovative probability assuming dimensions of the subspaces $\mathcal{SP}_i$ and $\mathcal{SP}_j$ are known. Generally, we define the *probability mass function* (PMF) of the dimensions of the subspaces for the *average peer* n as $f(k)$, and we can calculate the lower bound of the *average innovative probability*, P_{inv}^B, by a weighted average

$$P_{inv}^B = \sum_{k_i=1}^{B}\sum_{k_j=1}^{B} P_{inv}^{k_i,k_j,B} f(k_i)f(k_j). \tag{8}$$

B. *Innovative Probability for Multi-Stream*

When there are multiple streams being repaired simultaneously, our assumption that the components of the native coefficient vectors are randomly generated from $GF(O)$ is altered. This is because when a peer forwards a stream that he/she is not watching, he/she can only encode a packet using packets received from other peers through CPR without any packets received directly from WWAN. Without the chance of mixing the packets, the randomness of the components in the native coefficient vectors is reduced and thus our previous assumption does not hold.

To better understand the problem, let us consider a scenario where all peers are repairing two streams: s^1 and s^2. Assuming peers randomly select one stream to watch, then for a peer n watching stream s^1, half of n's neighbors are also watching s^1, and they can each send NC packets to n with innovative probability P_{inv}^B. The innovative probability of NC packets sent from the other half of n's neighbors to n, who are watching stream s^2, depends in turn on their neighbors, i.e., two-hop neighbors of n. Again, with probability 1/2, n's two-hop neighbors are watching s^1 and can help n via n's one-hop neighbors. For the rest half two-hop neighbors that watch s^2 can also receive some packets of stream s^1 during the repairing process, and with these limited packets they can help as well.

At this point, we need to consider the *common neighbor* effect where n's one-hop neighbors can receive identical packets from the same two-hop neighbor of n. Note we do not apply this effect to the two-hop neighbors who watch s^1 because different common one-hop neighbors may belong to many common neighbor groups and they can receive different packets from those two-hop neighbors during the CPR process, which greatly reduces the effect. However, this is not true for the two-hop neighbors who watch stream s^2 and have limited packets belonging to s^1.

The common neighbor effect is illustrated in Fig. 4, where peers n_5 and n_6 receive the same packet of stream s^1 from n_2, which reduces the innovative probability of subsequent NC packets forwarded to n by half. The innovative probability for the two stream scenario can now be estimated as $(1/2)P_{inv}^B + (1/2)\left(P_{inv}^B/4 + P_{inv}^B/2\right)$.

In general, denoting the average number of common neighbors as N_c, the average innovative probability for multi-stream is estimated as

$$\begin{aligned} P_{inv}^{B,S} &\approx \frac{1}{S}P_{inv}^B + \left(1-\frac{1}{S}\right)\left[\frac{P_{inv}^B}{N_c - \frac{N_c}{S}}\left(1-\frac{1}{S}\right) + P_{inv}^B\frac{1}{S}\right] \\ &= \left(2-\frac{1}{S}\right)\frac{P_{inv}^B}{S} + \left(1-\frac{1}{S}\right)\frac{P_{inv}^B}{N_c}. \end{aligned} \tag{9}$$

The first term in (9) accounts for neighbors watching the same stream as the receiving peer under consideration, and the second term accounts for neighbors watching different streams. Note that our derivation is limited to two-hop neighbors, which is conservative.

When SNC is considered, the innovative probability is estimated similarly as in the UNC case, except we set the batch size B to the size of the frame group that is under repair. Note that although we can get the simulated innovative probability under some scenarios offline, we cannot get it under all cases because in practice the topology of the network may change and N_c may change. In the following, we will use the analytical innovative probability for SNC optimization.

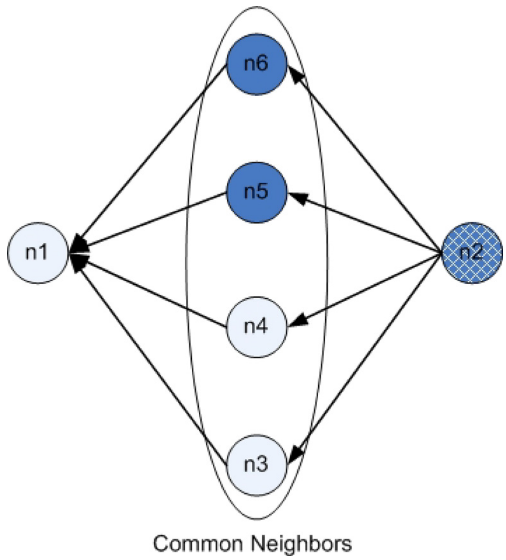

Fig. 4. Common neighbors in CPR network. n_3, n_4, n_5, and n_6 are common neighbors of n_1 and n_2. n_1, n_3, and n_4 watches s^1, and n_2, n_5, and n_6 watches s^2. n_2 receives one packet of s^1 during the repair process.

V. SNC Optimization Framework

In this section, we propose a framework to optimize structures and transmissions of network-coded CPR packets at peers so that the expected distortions of streams are minimized. Our proposed SNC optimization has two steps. First, the media source defines a global NC structure to minimize distortion for the average peer with average connectivity. Second, at each transmission opportunity a peer selects a stream from $\mathcal{A}_n$ and a type within the defined NC structure to transmit given its available local state information of its neighbors. We discuss the two steps in order.

A. Global NC Structure Definition

The media source first optimizes an NC structure for each stream s for the *average peer* n, assuming that an average peer can expect R_n packets from neighbors during CPR. Using the DAG source model from Section II-B, the expected distortion at peer n watching stream s can be written as

$$\Delta_n = D - \sum_{i=1}^{M} d_i \prod_{j \preceq i} \alpha_n(j). \tag{10}$$

d_i is calculated as the additional PSNR improvement of using decoded frame i for display of frame i, plus the PSNR improvement of using decoded frame i for error concealment of descendant frames of frame i in the source dependency tree in the event that they are incorrectly decoded, minus the PSNR improvement of using the parent frame of frame i (if one exists) for error concealment of frame i and its descendant frames. D is the sum of all d_i in one GOP, i.e., the distortion when no frame is received. $\alpha_n(j)$ is the recovery success probability of frame F_j at peer n. Note that in (10) we make the simplifying assumption that the frame recovery probability is independent from each other.

$\alpha_n(j)$ itself can be written as

$$\alpha_n(j) = (1-l)^{B_j} + \left(1 - (1-l)^{B_j}\right) S_n(j) \tag{11}$$

where l is the WWAN 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 WWAN.

Suppose we are given SNC groups $\Theta_1, \ldots, \Theta_X$. 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. We can hence write $S_n(j)$ as

$$S_n(j) \approx Q(n, g(j)) + \sum_{y=g(j)+1}^{X} Q(n,y) \prod_{z=g(j)+1}^{y} (1 - Q(n, z-1)) \tag{12}$$

where $Q(n,x)$ is the probability that peer n can NC-decode SNC type x by receiving $\sum_{F_i \in \Theta_x} B_i$ innovative native or NC packets. Note here we make the simplifying assumption that the recoveries of the frame groups are uncorrelated.

Using the average innovative probability shown in (8), if a peer n sends a NC packet of type x with probability $\beta_n(x)$, we can approximate $Q(n,x)$ as in (13) at the bottom of the page, where $L_j = \sum_{F_i \in \Theta_j} B_i$ is the number of packets in group j. $\Omega_x = \lceil l L_x \rceil$ is the expected number of lost packets of type x due to WWAN broadcast and needed CPR repairs. $1/S$ is the probability of receiving a particular stream given an active set of S streams. In words, (13) finds the frame group recovery probability by looking at the complimentary event that the frame group cannot be recovered, i.e., less than Ω_x innovative packets of SNC types $\leq x$ are received. Among the expected received R_n CPR packets, k of them are of SNC types $\leq x$ and are innovative. These packets are useful for n to recover frame group x. For the rest $R_n - k$ packets, some of them are of SNC types $\leq x$

$$Q(n,x) \approx 1 - \sum_{k=0}^{\Omega_x - 1} \binom{R_n}{k} \left(\frac{1}{S} \sum_{i=1}^{x} \beta_n(i) P_{inv}^{L_i,S} \right)^k \sum_{t=0}^{R_n-k} \binom{R_n-k}{t} \times \left(\frac{1}{S} \sum_{i=x+1}^{X} \beta_n(i) + \frac{S-1}{S} \right)^t \left(\frac{1}{S} \sum_{i=1}^{x} \beta_n(i) \left(1 - P_{inv}^{L_i,S} \right) \right)^{R_n-k-t} \tag{13}$$

but are not innovative; some of them are of SNC types greater than x. These packets are not useful for n to recover frame group x.

With our formulation shown in (10) —(13), the SNC optimization at the media source is 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 as follows:

$$\min_{X,\{\Theta_x\},\{\beta_n(x)\}} \Delta_n. \tag{14}$$

To solve the optimization problem in (14), a simple exhaustive search scheme has been shown to be of exponential complexity [12]. We therefore used an efficient local search algorithm for fast optimization.

We first notice that the search space can be reduced by considering the DAG structure described in Section II-B. 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 NC types then, we will assign preceding frames with a smaller or equal NC type than succeeding frames given the DAG structure.

Based on the reduced search space, we perform the local search as follows. We first assign M NC 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 NC type smaller than F_i. For this NC structure, we exhaustively search the best $\beta_n(x)$ resulting in the smallest distortion using (14). We then find the best "merging" of parent and child frames—assigning the same NC 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.

With our local search scheme, we need to check at most M merging operations for M frames in each iteration, and there are at most M iterations. Hence there are at most M^2 merge operations performed, which is significantly less than the exhaustive search. In practice, M is small, and by restricting the search space of $\beta_n(x)$ to 0.1—0.9 with 0.1 increment, we can bound the optimization in a reasonable amount of time, which facilitates real-time video streaming.

B. SNC Local Peer Optimization

1) Peers Utilize Local State Information: In the previous section, an NC structure was globally optimized for the entire ad-hoc network assuming an average peer with average connectivity. During CPR, however, *local state information* can be easily exchanged among neighbors by piggybacking on data packets with minimal overhead. By local we mean only one-hop neighbor information. Specifically, we assume each NC packet from peer n reveals which stream the packet is repairing and which stream n is watching [$\mathcal{S}(n)$]. The NC packet also includes two state reports: 1) *native packet reception report* identifying which packets of stream $\mathcal{S}(n)$ were successfully delivered from WWAN, and 2) *NC group status report* containing the number of innovative packets that are received in each NC groups of $\mathcal{S}(n)$. Note that the obtained local neighbor information can become inaccurate (stale) over time.

Using local information, a peer first selects a stream among $\mathcal{A}_n$ for repair deterministically instead of picking one at random. For a chosen stream, a peer then selects a NC packet type to transmit deterministically. This can potentially further improve streaming performance locally beyond the global optimization performed in previous section; for example, if a peer's neighbors have already fully recovered a certain stream, then the peer will not choose that stream for repair.

2) Local Peer Optimization: Using the local information discussed above, at each transmission opportunity a peer can select the optimal stream for repair and the SNC type that results in the minimum total distortion among all its neighbors. More specifically, we optimize the following expression:

$$\min_{v\in\mathcal{A}_n,u\in\mathcal{U}_n^v} \sum_{m\in\{n\text{'s neighbors}\}} \Delta_m^{v,u} \tag{15}$$

where v and u are the stream and the SNC type to be decided for packet transmission. $\mathcal{U}_n^v$ is the set of SNC types in stream v peer n has. Similar to (10), $\Delta_m^{v,u}$, the resulting distortion of neighbor m when NC packet of type u in stream v is transmitted, is written as

$$\Delta_m^{v,u} = D - \sum_{i=1}^{M} d_i \prod_{j\preceq i} \alpha_m^{v,u}(j). \tag{16}$$

Note here the distortion reduction is for neighbor m, and D, M, and d_i are constants for stream $\mathcal{S}(m)$. Since peer n has local information from neighbor m, we have

$$\alpha_m^{v,u}(j) = \begin{cases} 1, & \text{if frame } j \text{ has been received} \\ S_m^{v,u}(j), & \text{otherwise.} \end{cases} \tag{17}$$

Note that the first line in (17) has two meanings: either all the packets in frame j of stream $\mathcal{S}(m)$ are successfully delivered through WWAN or they have been repaired through CPR. They are inferred from the native packet reception report and the NC group status report, respectively. $S_m^{v,u}(j)$ has similar formulation as in the global NC definition part except here we need to decide the stream and packet type for transmission. It is now approximated as

$$Q^{v,u}(m,g(j)) + \sum_{y=g(j)+1}^{X} Q^{v,u}(m,y) \prod_{z=g(j)+1}^{y} \left(1 - Q^{v,u}(m,z-1)\right). \tag{18}$$

Since peers now have neighbor information, $Q^{v,u}(m,x)$ is updated as in (19) at the bottom of the next page, where $L_m^{x,v,u}$

is the number of innovative packets of type $\leq x$ peer m needs to recover frame group x, which can be written as

$$\begin{cases} C_m^x - R_m \frac{t}{Y}\frac{1}{S}\sum_{i=1}^{x}\beta_m(i)P_{inv}^{L_i,S} - P_{inv}^{L_x,S}, & v=\mathcal{S}(m), u=x \\ C_m^x - R_m\frac{t}{Y}\frac{1}{S}\sum_{i=1}^{x}\beta_m(i)P_{inv}^{L_i,S}, & \text{otherwise.} \end{cases} \quad (20)$$

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. $R_m(t/Y)(1/S)\sum_{i=1}^{x}\beta_m(i)P_{inv}^{L_i,S}$ represents the estimated number of innovative packets of type $\leq x$ in stream $\mathcal{S}(m)$ neighbor m could receive during time interval t. If the transmitted stream v is the same as the stream peer m needs, $\mathcal{S}(m)$, and the transmitted packet type u is the same as x, then the packet transmitted from n to m will be an innovative packet with probability $P_{inv}^{L_x,S}$, which results in a reduction in the needed number of packets. Similarly, U_m is the total number of packets neighbor m could possibly receive during the rest of the repair time. It is written as

$$U_m = \left\lfloor R_m\left(1 - \frac{t'}{Y}\right)\right\rfloor - 1 \quad (21)$$

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

Note that in (19) and (20), we assume conservatively that peer m's other neighbors do not perform local optimization, but instead are transmitting using the predetermined transmission probability. This is due to the fact that to predict the optimization results of peer 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 difficult to achieve in a distributed scenario.

VI. Simulation Studies

In this section, we verify the effectiveness of our SNC optimization framework through simulations. We first present the simulation setup: the video codec parameters and the CPR network settings. Next, we show the result of the innovative probability estimation. We then compare the performance of the UNC and SNC schemes when CPR bandwidth is not sufficient to repair all WWAN losses for each stream. Finally, we examine the benefits of the two proposed innovations in our SNC framework: local peer optimization and innovative probability estimation.

A. Simulation Setup

Two test video sequences were used for simulations: 300-frame MPEG class A `news` and class B `foreman` sequences at QCIF resolution (176×144), at 30 fps and sub-sampled in time by 2. The GOP size was chosen at 15 frames: one I-frame followed by 14 P-frames. Quantization parameters used for I-frames and P-frames were 30 and 25, respectively. The H.264 codec used was JM 12.4, downloadable from [20]. We performed reference frame selection in [15] with target encoding rate at 220 kbps, resulting in a DAG describing inter-frame dependencies as discussed in Section II-B. For each trial, we used the same video sequence as media content for all streams. A peer selected a stream to watch randomly among all available streams.

We considered a CPR network of size $1000 \times 1000\ \mathrm{m}^2$ where 50 peers were uniformly distributed. The peers were watching video streams through MBMS using their multi-homed devices, where WLAN interfaces were activated for CPR. We used the broadcast mode of WLAN, therefore no feedback messages were sent from the receivers and no transmission rate adaption was performed. The media source provided S_{all} streams, each of which was transmitted at rate $r_{\mathrm{MBMS}} = 220$ kbps. Given one GOP was 15 frames and video was encoded at 15 fps, one epoch time Y is 1 s. The MBMS broadcast packet loss rate was kept constant at 0.1. Each CPR packet is set to the size $W = 1000$ bytes. We used QualNet [21] to conduct the simulations. To have the freedom to vary CPR bandwidth, we selected *Abstract PHY* in QualNet for physical layer and set all of the parameters to be the default values in 802.11.

B. Simulation Results

1) Innovative Probability: We compared our analytical results on innovative probability to the simulation results in this section. Simulations for both the single stream and multi-stream scenarios were performed. The video sequence in use was the `news` sequence. The CPR bandwidth was 4.5 Mbps, which is the typical data rate for 802.11b.

Fig. 5(a) plots the average innovative probability when all the peers were watching the same stream and used UNC scheme to do the repairing. Since the average number of initial packet loss was lB, where l is MBMS packet loss rate, we assumed that PMF $f(k)$ was uniformly distributed between $(1-l)B$ and B. This assumption is reasonable because during the repairing process, the dimensions of the encoding coefficient vectors were increasing gradually and steadily. Because of the

$$Q^{v,u}(m,x) \approx 1 - \sum_{k=0}^{\lceil L_m^{x,v,u}\rceil - 1}\binom{U_m}{k}\left(\frac{1}{S}\sum_{i=1}^{x}\beta_m(i)P_{inv}^{L_i,S}\right)^k \sum_{t=0}^{U_m-k}\binom{U_m-k}{t} \times \left(\frac{1}{S}\sum_{i=x+1}^{X_m}\beta_m(i) + \frac{S-1}{S}\right)^t \left(\frac{1}{S}\sum_{i=1}^{x}\beta_m(i)\left(1 - P_{inv}^{L_i,S}\right)\right)^{U_m-k-t} \quad (19)$$

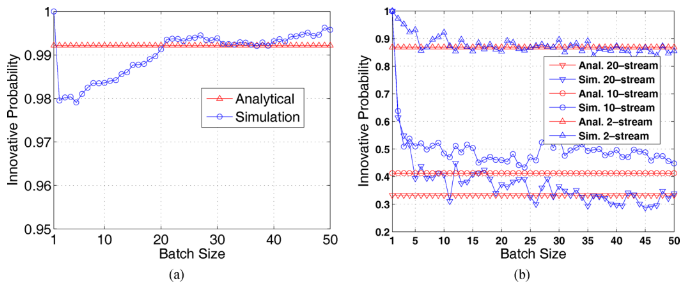

Fig. 5. Receiving CPR packet innovative probability. a) Single stream. b) Multi-stream.

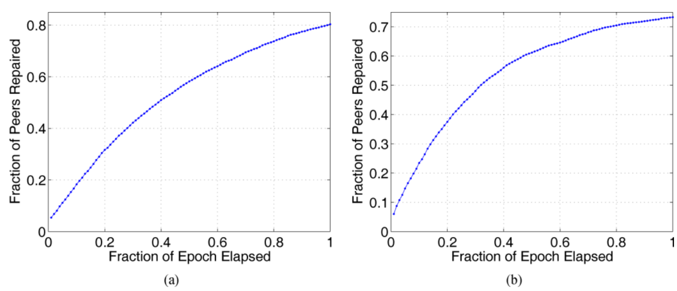

Fig. 6. CDF of the number of peers repaired during one epoch time. a) CPR BW 4.5 Mbps, $S = 10$. b) CPR BW 23 Mbps, $S = 20$.

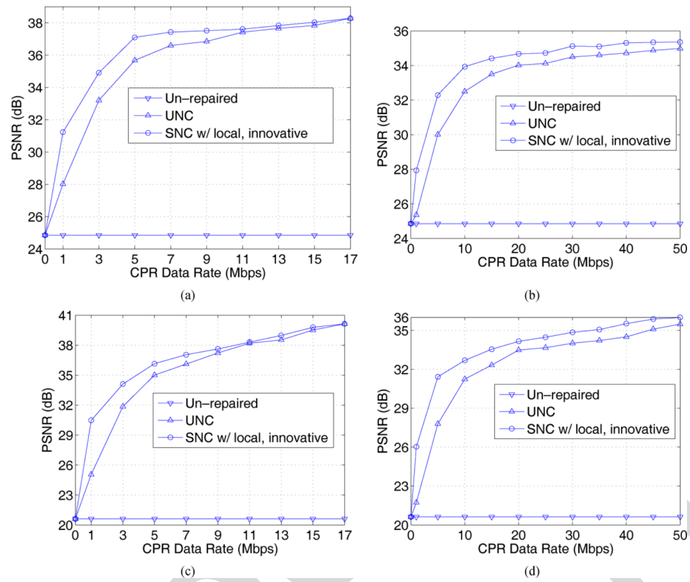

Fig. 7. PSNR for `news` and `foreman` under various CPR data rates. a) `news` ten streams. b) `news` 20 streams. c) `foreman` ten streams. d) `foreman` 20 streams.

low packet loss rate, peers received most of the packets from MBMS. Therefore each transmitted NC packet is a combination of a large number of native and NC packets, which makes the components of the native coefficient vectors random and the innovative probability close to 1. The difference between the analytical and simulation results was small and was due to the simplified assumption of uniform distribution on the dimension of subspaces.

Fig. 5(b) shows the analytical result versus the simulation result under various multi-stream scenarios. Intuitively, with the increase of the number of video streams, the innovative probability is reduced. We see that the analytical results capture the trend of the simulation results very well.

2) Multi-Stream Repair With UNC: As discussed in Section III, if a peer does not receive a sufficient number of innovative native or NC packets during CPR to recover *all* WWAN losses, then UNC could not recover *any* lost packets using received NC packets. This undesired phenomenon was depicted in Fig. 6(a), which shows the CDF of the fraction of peers that recovered all packets through CPR in one epoch time using UNC. There were $S = 10$ total active streams, and on average 5 peers were watching the same stream. CPR operated at the typical 802.11b data rate. As shown, only about 80% of peers recovered their lost packets in one epoch time. Similarly, Fig. 6(b) shows the CDF when there were $S = 20$ total active streams, and the CPR bandwidth was increased to 23 Mbps, the typical data rate for 802.11a/g. The result was similar, and fewer than 75% of the peers benefited from CPR with UNC.

3) Multi-Stream Repair With SNC: We now show the performance of SNC for the multi-stream scenario. The complete SNC scheme involves a two-step optimization: 1) media source first searches for the optimal NC structure for each stream separately using the optimization framework shown in Section V; and 2) individual peer performs local optimization by utilizing partial state information received from neighbors. When a peer has received enough packets for a certain frame group, the packets within that particular frame group can be recovered. With our SNC frame group optimization, it turned out that when the CPR bandwidth was low, the SNC optimization returned more NC types than when the bandwidth was high. We also noted that the lower the bandwidth was, the smaller the sizes of the first few NC groups. This is reasonable because when bandwidth is low, peers need desperately to decode at least the first few frames. Dividing the packets into more groups increases the chance that the received packets can be decoded, and therefore peers can at least decrease some of the distortion with the limited number of receiving packets.

In the following, we first compare the performance of SNC to UNC under different CPR data rates using different video sequences. We then show the effectiveness of the local peer optimization and the innovative probability estimation in the SNC optimization framework. Lastly we explore how the number of streams affected the performance.

SNC Outperforms UNC: Fig. 7(a) and (b) shows the CPR data rates versus PSNR plot for `news` when there were ten and 20 streams, respectively. Fig. 7(c) and (d) shows the CPR data rates versus PSNR plot for `foreman`. We also have the un-repaired video quality, the original video quality without any CPR repairs, as a performance benchmark.

From Fig. 7 it can be easily observed that SNC outperformed traditional UNC and un-repaired video in all transmission rates. When there were ten streams provided by MBMS, SNC provided up to 13.51 dB PSNR improvement for the `news` sequence and 19.71 dB PSNR improvement for the `foreman` sequence over un-repaired video when the data rate was larger than 17 Mbps. When there were 20 streams, the performance improvement over un-repaired video using SNC were up to 10.51 dB and 15.37 dB when the data rate was larger than 50 Mbps. For UNC, the peers needed $\sum_{j=1}^{15} B_j$ innovative native or NC packets before any repairing could be performed. However, for the SNC scheme, peers could repair important frames as soon as sufficient NC packets of particular SNC types were received. Hence when bandwidth was low, the performance of SNC was much better than UNC. For example, at the transmission rate of 1 Mbps, SNC achieved 3.21 dB gain over UNC for the `news` sequence and around 5.39 dB gain for the `foreman` sequence where there were ten streams. When the bandwidth was higher, the number of received packets increased so that UNC recovered more packets and the performance of the two schemes became similar. Note that when there were ten streams, when the 802.11 data rate exceeded 17 Mbps, all the packets could be repaired for both `news` and `foreman`. However when there were 20 streams, even when the 802.11 data rate was almost at maximum, 50 Mbps, there were still packet loss. Therefore it is always better to choose SNC over UNC when the number of streams is large. We note that with the increase of CPR data rate, the slopes of the curves were reducing. We explain this phenomenon with following three reasons: 1) with the increase of CPR data rate, the packet loss rate was also increased, which reduced the effective bandwidth; 2) distortions of the frames in a GOP was not uniformly distributed. With the first few received packets, more distortion could be recovered through CPR; 3) the packet innovative probability reduced with the increased number of receiving packets.

Comparing the video qualities for the `news` and `foreman` sequences, we found that the improvement by using SNC over the UNC scheme was more pronounced for the `foreman` sequence. For example, as shown earlier the gain was 3.21 dB for the `news` sequence and 5.39 dB for the `foreman` sequence when ten streams were repaired under 1 Mbps CPR data rate. This is due to the fact that `foreman` has more inherent motion and requires more encoding bits for the same given quantization parameters. As a result, the corresponding DAG was long rather than wide, which means that if a particular packet close to the root node is lost, it affects many descendant frames and results in large distortion.

Effectiveness of Local Peer Optimization and Innovative Probability Estimation: We also examine the individual benefits of the two innovations we propose within the SNC framework: local peer optimization and innovative probability estimation. We compare the performance when: 1) both innovations were removed; 2) only innovative probability estimation was added; and 3) both innovations were added.

Fig. 8(a) and (b) compares the performance of SNC under different configurations for both the `news` and `foreman` sequences. First, note that SNC without both innovations already outperformed UNC for all configurations. For example at 1 Mbps CPR data rate, for the `news` sequence and without local optimization and innovative probability estimation (innovative probability set to 1), SNC achieved a gain of 1.54 dB over UNC. When we used innovative probability estimation only, we reaped 2.65 dB gain over the UNC scheme. By utilizing both local peer optimization and innovative probability estimation, SNC provided 3.21 dB gain over UNC. The results were similar for the `foreman` sequence.

Number of Streams Affects Performance: Fig. 9 shows the performance of UNC and SNC when the stream number varied

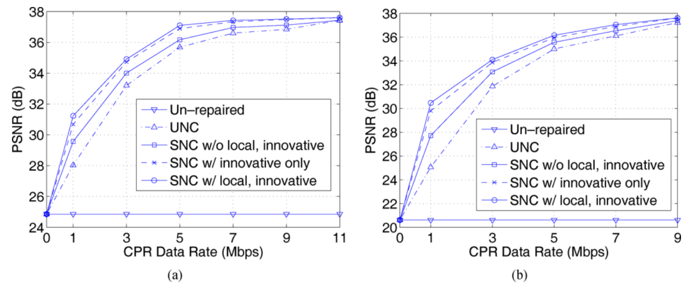

Fig. 8. PSNR for the `news` and `foreman` sequences under various CPR transmission rates and SNC scheme settings. a) `news` ten streams. b) `foreman` ten streams.

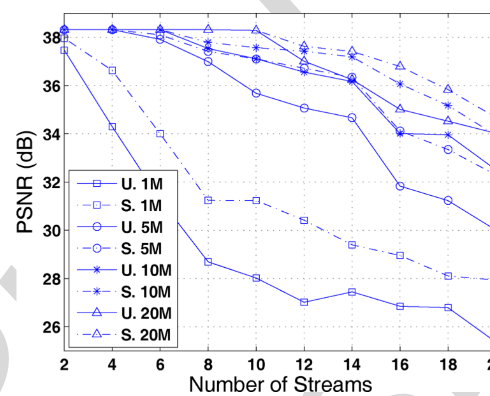

Fig. 9. PSNR for the `news` sequence under various multi-stream scenarios. U. and S. are short for UNC and SNC, respectively.

from 2 to 20. Obviously with the increase of the number of video streams, performance decreased because the CPR bandwidth that could be allocated to a particular stream was reduced. Peers had to contribute most of their CPR bandwidth to help others. Nevertheless, our SNC scheme showed noticeable gain over the UNC scheme for all cases.

VII. Related Work

Due to the aforementioned NAK implosion problem [3], many video streaming strategies over MBMS [4] have forgone feedback-based error recovery schemes like [2] and opted instead for forward error correction (FEC) schemes like Raptor codes [4]. 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. We instead exploit the multi-homed nature and propose to repair lost packets through CPR.

NC has been a popular research area since Ahlswede's seminal work [22], which showed that network capacity can generally be achieved using NC. Many studies have since explored message dissemination using NC. In [23], the authors proposed to use random NC [17] to encode the packets to be transmitted in a peer-to-peer content delivery scenario. We leverage this idea to our design and focus on video streaming and NC structure in wireless ad-hoc networks. A gossip-based protocol was proposed in [19] which utilizes network coding to disseminate messages. Instead of gossiping, we utilize the broadcast nature of the wireless medium to disseminate video packets.

Recent works [18], [24]–[26] have attempted to jointly optimize video streaming and NC. [18] 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 GOP, while [18] is performed greedily per packet.

Reference [24] utilized the hierarchical NC scheme in the same way for CDN and P2P networks to combat Internet bandwidth fluctuation. Our work is more general in that our source model is a DAG, while the model in [24] is a more restricted dependency chain. Moreover, we provide a NC optimization framework to better exploit the benefit of SNC.

[25] discussed the application of Markov Decision Process [16] to NC, in which NC optimization and scheduling are centralized at the access point or base station. Like [18] 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 work, we instead consider fully distributed peer-to-peer repair without assuming full knowledge of state information of peers.

Reference [26] discussed applying structure on NC across multiple generations of video packets, where one generation is defined at the transport layer irrespective of application-layer GOP structures. In our work, NC is applied within one GOP, and the structure is defined according to the dependency tree among the video frames in the GOP. Defining NC structure within a GOP enables us to build a rate-distortion based NC optimization framework which finds the optimal NC structure resulting in the smallest expected distortion. To our knowledge, we are also the first in the NC literature to use randomization in the implementation of SNC for video streaming optimization.

VIII. Conclusions

In this paper, we present a novel, rate-distortion optimized, NC-based, cooperative peer-to-peer packet repair solution for the multi-stream WWAN video broadcast. We make contributions in the following major aspects. First, we propose a two-step NC structure optimization framework in which the video stream repair can be optimized in a rate-distortion manner. Second, we analyze the innovative probability of a receiving NC packet to facilitate accurate NC structure optimization. Lastly, we provide detailed simulations and show that the video quality can be improved by up to 19.71 dB over un-repaired video stream and by up to 5.39 dB over video stream using traditional unstructured network coding.

References

[1] 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, 2006.

[2] G. Cheung, W.-T. Tan, and T. Yoshimura, "Double feedback streaming agent for real-time delivery of media over 3G wireless networks," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 304–314, Apr. 2004.

[3] J. Crowcroft and K. Paliwoda, "A multicast transport protocol," in *Proc. ACM SIGCOMM*, New York, Aug. 1988.

[4] J. Zfzal, T. Stockhammer, T. Gasiba, and W. Xu, "Video streaming over MBMS: A system design approach," *J. Multimedia*, vol. 1, no. 5, pp. 25–35, Aug. 2006.

[5] P. Sharma, S.-J. Lee, J. Brassil, and K. Shin, "Distributed communication paradigm for wireless community networks," in *Proc. IEEE Int. Conf. Communications*, Seoul, Korea, May 2005.

[6] S. Raza, D. Li, C.-N. Chuah, and G. Cheung, "Cooperative peer-to-peer repair for wireless multimedia broadcast," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, Beijing, China, Jul. 2007, pp. 1075–1078.

[7] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, "Network coding-based broadcast in mobile ad-hoc networks," in *Proc. IEEE INFOCOM 2007*, Anchorage, AL, May 2007, pp. 1739–1747.

[8] X. Liu, S. Raza, C.-N. Chuah, and G. Cheung, "Network coding based cooperative peer-to-peer repair in wireless ad-hoc networks," in *Proc. IEEE Int. Conf. Communications (ICC)*, Beijing, China, May 2008.

[9] A. Rahmati and L. Zhong, "Context-for-wireless: Context-sensitive energy-efficient wireless data transfer," in *Proc. ACM MobiSys*, San Juan, PR, Jun. 2007, pp. 165–178.

[10] S. Chandra and A. Vahdat, "Application-specific network management for energy-aware streaming of popular multimedia formats," in *Proc. USENIX Annu. Conf.*, Monterey, CA, Jun. 2002, pp. 329–342.

[11] T. Pering, Y. Agarwal, R. Gupta, and R. Want, "Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces," in *Proc. ACM MobiSys*, Jun. 2006, pp. 220–232.

[12] 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 *Proc. 1st IEEE Workshop on Mobile Video Delivery*, Newport Beach, CA, Jun. 2008.

[13] X. Liu, G. Cheung, and C.-N. Chuah, "Structured network coding and cooperative local peer-to-peer repair for MBMS video streaming," in *Proc. Int. Workshop on Multimedia Signal Processing*, Cairns, Queensland, Australia, Oct. 2008.

[14] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[15] G. Cheung, W.-T. Tan, and C. Chan, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 649–662, Jun. 2007.

[16] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[17] 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 Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[18] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," in *Proc. IEEE 16th Int. Packet Video Workshop*, Lausanne, Switzerland, Nov. 2007, pp. 191–200.

[19] S. Deb, M. Medard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2486–2507, Jun. 2006.

[20] The TML Project Web-Page and Archive. [Online]. Available: http://iphome.hhi.de/suehring/tml/.

[21] Qualnet. [Online]. Available: http://www.scalable-networks.com/.

[22] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeun, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[23] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE Infocom*, Mar. 2005, pp. 2235–2245.

[24] K. Nguyen, T. Nguyen, and S.-C. Cheung, "Video streaming with network coding," *Springer J. Signal Process. Syst Special Issue: ICME07*, Feb. 2008.

[25] D. Nguyen, T. Nguyen, and X. Yang, "Multimedia wireless transmission with network coding," in *Proc. IEEE 16th Int. Packet Video Workshop*, Lausanne, Switzerland, Nov. 2007, pp. 326–335.

[26] M. Halloush and H. Radha, "Network coding with multi-generation mixing: Analysis and applications for video communication," in *Proc. IEEE Int. Conf. Communications*, May 2008.

Xin Liu (S'XX) **AUTHOR: MISSING MEMBER YEARS.** received the B.S. degree in electronic and information engineering and the M.E. degree in computer engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2003 and 2005, respectively. In 2006, he received the M.S. degree in electrical engineering from the State University of New York at Buffalo, Buffalo, NY. He is pursuing the Ph.D. degree in computer engineering at the University of California, Davis.

His research interests are wireless networking, cooperative communications, and network coding.

Gene Cheung (SM'XX) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1995 and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1998 and 2000, respectively.

In August 2000, he joined Hewlett-Packard Laboratories Japan, Tokyo, where he is currently a Senior Researcher. His research interests include media transport over wireless networks, joint source/network coding for video streaming, and community-based media interaction.

Dr. Cheung is currently an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and a voting member of the IEEE Multimedia Communications Technical Committee.

Chen-Nee Chuah (SM'XX) received the B.S. degree in electrical engineering from Rutgers University, New Brunswick, NJ, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley.

She is currently an Associate Professor in the Electrical and Computer Engineering Department at the University of California, Davis. Her research interests include Internet measurements, network management, and wireless/mobile computing.

Dr. Chuah received the NSF CAREER Award in 2003 and the Outstanding Junior Faculty Award from the UC Davis College of Engineering in 2004. In 2008, she was selected as a Chancellor's Fellow of UC Davis. She has served on the executive/technical program committee of several ACM and IEEE conferences and is currently an Associate Editor for IEEE/ACM TRANSACTIONS ON NETWORKING.

Structured Network Coding and Cooperative Wireless Ad-Hoc Peer-to-Peer Repair for WWAN Video Broadcast

Xin Liu, *Student Member, IEEE*, Gene Cheung, *Senior Member, IEEE*, and Chen-Nee Chuah, *Senior Member, IEEE*

***Abstract*—In a scenario where each peer of an ad-hoc wireless local area network (WLAN) receives one of many available video streams from a wireless wide area network (WWAN), we propose a network-coding-based cooperative repair framework for the ad-hoc peer group to improve broadcast video quality during channel losses. Specifically, we first impose network coding structures globally, and then select the appropriate video streams and network coding types within the structures locally, so that repair can be optimized for broadcast video in a rate-distortion manner. Innovative probability—the likelihood that a repair packet is useful in data recovery to a receiving peer—is analyzed in this setting for accurate optimization of the network codes. Our simulation results show that by using our framework, video quality can be improved by up to 19.71 dB over un-repaired video stream and by up to 5.39 dB over video stream using traditional unstructured network coding.**

***Index Terms*—Cooperative peer-to-peer repair, network coding, wireless wide area network (WWAN) video broadcast.**

I. Introduction

With consumers' increasing demand for rich media contents and the ubiquity of mobile wireless access, deployments of various wireless multimedia services are fast emerging. To scale these services to large user bases, different wireless wide area network (WWAN) multimedia broadcast/multicast technologies have been proposed. For example, Multimedia Broadcast/Multicast Service (MBMS) [1] was introduced in UMTS cellular networks of 3GPP release 6.0 and later, which provides efficient point-to-multipoint multimedia delivery via a common cellular channel.

While the broadcast nature of the aforementioned WWAN multimedia distribution technologies enables scalable and bandwidth-efficient media delivery to a larger number of users via a common physical channel, it also has its share of technical challenges. First, previously developed feedback-based loss recovery schemes like [2] for point-to-point unicast streaming become infeasible in the broadcast scenario due to either the lack of a feedback channel, or the well-known NAK implosion problem [3] even if such feedback channel is available. Second, because broadcast systems are often optimized for the average channel [4] to maximize utility for the average user, packet losses are inevitable for the temporarily-worse-than-average users due to the unpredictable and time-varying nature of wireless channels, resulting in deteriorated video quality.

Given the recent popularity of multi-homed mobile devices [5]—devices with both 3G cellular and IEEE 802.11 wireless interfaces—one potential solution to the broadcast packet loss problem is for a group of interconnected peers listening to the same video stream to use their 802.11 interfaces to cooperatively perform *out-of-band* repair of 3G broadcast losses. This is the premise behind our previously proposed *cooperative peer-to-peer repair* (CPR) framework [6] to combat WWAN packet losses. Having each correctly received a different subset of packets from WWAN 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 WWAN packets. Using our developed heuristics, we showed in [6] that significant packet recovery can be achieved. Moreover, if we permit each peer to perform *network coding* (NC) [7]—linearly combining payloads of received packets in Galois Field $GF(O)$ where $O = 2^q$ is the field size and q is a positive integer—before forwarding packets, we showed in [8] that even further performance gain can be achieved.

Compared to its cellular counterpart, an 802.11 interface requires much more power to establish and maintain connections [9]–[11], and as a result, having both 3G and 802.11 interfaces activated constantly may not be feasible for lightweight battery-powered handheld devices consuming lengthy videos. To address the power consumption issue, we have previously imposed structures on NC [12], [13] to optimize repaired video quality given an energy budget.

In our previous works, we assumed that all peers in the same ad-hoc network are watching the same video; i.e., all available 802.11 bandwidth can be used to repair a single video stream. In practice, however, different users are likely watching different streams, and as a result, multiple streams (multi-stream) need CPR to improve broadcast video simultaneously. Fig. 1 illustrates the multi-stream scenario where different peers are watching different streams a, b, and c. Since each peer now needs to relay CPR packets of streams they are not watching, the network resource allocated to each stream is reduced. In this paper, we address this more realistic and more challenging scenario.

Manuscript received August 25, 2008; revised January 08, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. James E. Fowler.

X. Liu and C.-N. Chuah are with the Department of Electrical and Computer Engineering, University of California, Davis, Davis, CA 95616-5294 USA (e-mail: xyzliu@ucdavis.edu; chuah@ucdavis.edu).

G. Cheung is with Hewlett-Packard Laboratories Japan, Tokyo 168-0072, Japan (e-mail: gene-cs.cheung@hp.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2009.2017636

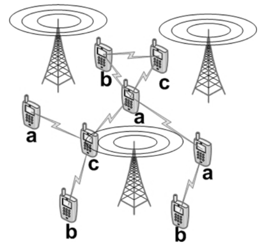

Fig. 1. Illustration of multi-stream scenario cooperative peer-to-peer repair.

Specifically, we present a rate-distortion optimized, NC-based, CPR solution for the multi-stream scenario to improve WWAN broadcast video quality. Our contributions are the following.

1) We propose a two-step NC optimization framework: 1) *global NC structure optimization*, where the media source defines an optimal NC structure globally based on the source's estimated average peer's network state, so that packets of more important frames can be recovered with appropriately higher probabilities for the average peer; 2) *local peer optimization*, where at a peer's transmission opportunity, given its available local state information at hand about its neighbors, a peer selects a stream and a NC type for packet transmission to minimize distortions particularly for its neighbors.
2) To facilitate accurate NC optimization, we estimate the *innovative probability* —likelihood that a received packet at a peer is useful for data recovery—in a computation-efficient manner.
3) We provided detailed simulations to verify our results, showing that our solution improves video quality significantly: by up to 19.71 dB over un-repaired video stream and by up to 5.39 dB over video stream using traditional unstructured NC schemes.

The outline of the paper is as follows. In Section II, we discuss the multi-stream system and our chosen source and network models. In Section III, we formally define unstructured NC and our proposed structured NC. In Section IV, we analyze packet innovativeness of receiving CPR packets at a given peer. Based on these discussions, we present our NC optimization framework in Section V. We explain our results in Section VI. We overview related works in Section VII and conclude in Section VIII, respectively.

II. System Architecture and Models

We first outline the architecture of our proposed broadcast video repair system. We then introduce two theoretical models used in our system optimization: 1) a video source model we use to optimize network coding for packet recovery, and 2) a network model used to schedule peer-to-peer packet repairs.

A. CPR System Architecture

We consider the scenario where N peers are watching broadcast video streams using their wireless mobile devices through the WWAN. The mobile devices are also equipped with *wireless local area network* (WLAN) interfaces, and the peers are physically located in close enough proximity that a peer-to-peer wireless ad-hoc network can be formed. The video streams can be live or stored content that are broadcasted from the *media source*; for simplicity, we denote media source to mean both a *media encoder* (where the video streams are encoded), and the actual video broadcasting entity over WWAN.

We first assume that the media source provides a total of S_{all} video streams. S_{all} varies due to different technologies, broadcast bandwidths, and operational constraints of the mobile video providers. Although S_{all} streams are available, not all streams will have audiences in a given ad-hoc network at a given time. Without loss of generality, we denote $\mathcal{S}^* = \{s^1, s^2, \ldots, s^S\}$ as the subset of S_{all} streams that have audience and $S = |\mathcal{S}^*|$. We assume that the media source can estimate the size S of the subset (rather than the actual subset $\mathcal{S}^*$ itself) based on its past history and inform the peers of its estimate using predefined fields in data packet headers.

Each peer n in the network watches one stream $\mathcal{S}(n) \in \mathcal{S}^*$ from the media source, and conversely each stream $s \in \mathcal{S}^*$ has a group of receiving peers $\mathcal{U}_s$. Peers in $\mathcal{U}_s$, each receiving a different subset of packets of stream s, can relay packets to others using WLAN interfaces to repair lost packets. This repair process is called CPR.

We assume that each peer is willing to relay repair packets of other streams; in return other peers will relay repair packets for the peer. We denote $\mathcal{A}_n$ as the set of streams of which peer n has received packets: either original video packets from the media source or CPR packets from peers, i.e., streams that peer n can repair via CPR. We use flags in CPR packet header to identify the stream a packet repairs. Whenever peer n has a transmission opportunity—a moment in time when peer n is permitted by a scheduling protocol (to be discussed) to locally broadcast a packet via WLAN, peer n selects one stream from $\mathcal{A}_n$ to construct and transmit a CPR packet.

B. Source Model

We use H.264 [14] codec for video source encoding because of its excellent rate-distortion performance. For improved error resilience, we assume the media source first performs *reference frame selection*[15] for each *group of picture* (GOP) in each stream separately during H.264 encoding. In brief, [15] assumes each GOP is composed of a starting I-frame followed by P-frames. Each P-frame can choose among a set of previous frames for *motion compensation*, where each choice results in a different encoding rate and different dependency structure. If we then assume that a frame is correctly decoded only if it is correctly received and the frame it referenced is correctly decoded, then this choice leads to a different correctly decoded probability. Using P-frames' selection of reference frames, [15] sought to maximize the expected number of correctly decoded frames given an encoding rate constraint.

After the media source performs reference frame selection for each GOP of each stream, we can model M^k frames in a GOP of a stream s^k, $\mathcal{F}^k = \{F_1^k, \ldots, F_{M^k}^k\}$, as nodes in a *directed acyclic graph* (DAG) as shown in Fig. 2, similarly done in [16]. Each frame F_i^k has an associated d_i^k, the resulting distortion reduction if F_i^k is correctly decoded. Each frame F_i^k points to the frame in the same GOP that it uses for motion compensation. Frame F_i^k referencing frame F_j^k results in encoding rate $r_{i,j}^k$.

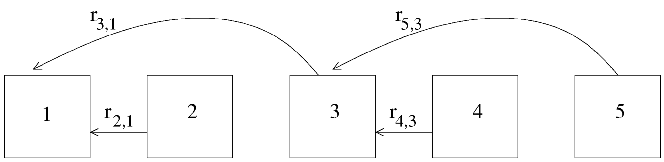

Fig. 2. Example of DAG source model for H.264/AVC video with reference frame selection.

We assume each frame F_j^k is packetized into *real-time transport protocol* (RTP) packets according to the frame size and *maximum transport unit* (MTU) of the delivery network. A frame F_j^k is correctly received only if all packets within F_j^k are correctly received.

We assume that the media source delivers each GOP of M^k frames of stream s^k in time duration Y^k. Y^k is also the *repair epoch* for s^k, which is the duration in which CPR completes its repair on the previous GOP; i.e., peers exchange CPR packets for previous GOP of stream s^k during the current epoch. The playback buffer delay for peer n is hence two epochs. Given that our later discussion focuses on one stream s^k, for simplicity we drop the superscript and refer to frame F_i^k simply as F_i, etc.

C. Network Model

As done in [8], we assume the multi-homed devices of N ad-hoc peers watching WWAN video perform CPR in 802.11 broadcast mode, so that a transmitted WLAN packet can potentially be received by more than one neighbor. Note that though raw WLAN transmission rate like 802.11 is relatively large, peers need to contend for the shared medium for transmission in a distributed manner so that the occurrences of collision and interference are reduced. For brevity, we omit the discussion on a distributed algorithm [8] that schedules WLAN ad-hoc peer transmissions. We simply assume that the average peer n can receive R_n total repair packets successfully via CPR in one repair epoch, which varies depending on available WLAN resources for CPR (constrained by factors such as power [12], [13] and contending cross traffic).

III. Network Coding Based CPR

In this section, we first describe *unstructured network coding* (UNC), common in the literature, in the context of CPR. We then present *structured network coding* (SNC), a new technique where by imposing structures on NC, one can further optimize NC specifically for video streaming in a rate-distortion manner.

A. Unstructured Network Coding

We denote the traditional random NC scheme [17] as UNC, as compared to our proposed SNC. First, suppose peer n has a transmission opportunity and n selects stream s from $\mathcal{A}_n$ for transmission. Suppose there are M original (native) frames $\mathcal{F} = \{F_1, \ldots, F_M\}$ in a GOP of stream s to be repaired among peers in $\mathcal{U}_s$. Each frame F_i is divided into multiple packets $\mathcal{P}_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,B_i}\}$ of size W bits each. Here B_i is the number of packets frame F_i is divided into. Note that a peer adds padding bits to each packet so that each has constant size W bits; this is performed for NC purposes, similarly done in [18]. We denote $\mathcal{P}^*$ as the set of all packets in a GOP, i.e., $\mathcal{P}^* = \{\mathcal{P}_1, \ldots, \mathcal{P}_M\}$. There are a total of $P = |\mathcal{P}^*| = \sum_{i=1}^{M} B_i$ packets to be disseminated among peers in $\mathcal{U}_s$.

We denote $\mathcal{G}_n$ as the set of *native packets* of stream $\mathcal{S}(n)$ peer n received from media source. Denote $\mathcal{Q}_n$ as the set of *NC packets* of stream s peer n received from other peers through CPR. If the stream selected for transmission is the same as the stream peer n currently watches, i.e., $s = \mathcal{S}(n)$, then the NC packet q_n generated by peer n is represented as

$$q_n = \sum_{p_{i,j} \in \mathcal{G}_n} a_{i,j} p_{i,j} + \sum_{q_m \in \mathcal{Q}_n} b_m q_m = \sum_{p_{i,j} \in \mathcal{P}^*} c_{i,j} p_{i,j} \quad (1)$$

where $a_{i,j}$'s and b_m's, random numbers in $GF(O)$, are coefficients for the original packets and the received encoded NC packets, respectively. Because each received NC packet q_m is itself a linear combination of native and NC packets, we can rewrite q_n as a linear combination of native packets with *native coefficients* $c_{i,j}$'s as shown in (1).

If the stream selected for transmission $s \neq \mathcal{S}(n)$, then the NC packet is simply a linear combination of all NC packets of stream s received through CPR from other peers so far as follows:

$$q_n = \sum_{q_m \in \mathcal{Q}_n} b_m q_m = \sum_{p_{i,j} \in \mathcal{P}^*} c_{i,j} p_{i,j}. \quad (2)$$

For UNC, *all* packets of stream s, both native packets (if any) and received NC packets, are used for NC encoding, and a peer in $\mathcal{U}_s$ can reconstruct all P native packets of stream s when P innovative native or NC packets of stream s are received, and hence all frames can be recovered. By innovative, we mean that native coefficient vector $\mathbf{v} = [c_{1,1}, \ldots, c_{1,B_1}, \ldots, c_{M,1}, \ldots, c_{M,B_M}]$ of a newly received packet is not a linear combination of native coefficient vectors from the set of previously received innovative packets. When a peer has accumulated P innovative packets, it recovers all P native packets in the GOP by solving P linear equations, each equation corresponding to an innovative packet, itself a sum of native packets as shown in (1).

The downside of UNC is that if a peer n receives fewer than P innovative packets, this 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. This is indeed the case for multi-stream, where the CPR bandwidth is shared by all streams, as we will see in Section VI. Hence there is a need to derive an alternative NC strategy for multi-stream.

B. Structured Network Coding

To address the aforementioned issue, we propose to use 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 of stream s are received. We accomplish that by forcing some chosen coefficients $a_{i,j}$'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) so that a subset of video frames in a GOP can be recovered.

More precisely, given the DAG source model described in Section II-B, for stream s, we first define a series of X *SNC frame groups*, $\Theta_1, \ldots, \Theta_X$, where $\Theta_1 \subset \cdots \subset \Theta_X = \mathcal{F}$. 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 as follows:

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

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 $\Phi(q)$. Similar to UNC, when the stream selected for transmission is the same as the stream that peer n watches, i.e., $s = \mathcal{S}(n)$, then the 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 $\mathcal{Q}_n$ is written as

$$\begin{aligned} q_n(x) = &\sum_{p_{i,j} \in \mathcal{G}_n} U(g(j) \le x)\, a_{i,j} p_{i,j} \\ &+ \sum_{q_m \in \mathcal{Q}_n} U(\Phi(q_m) \le x)\, b_m q_m = \sum_{p_{i,j} \in \Theta_x} c_{i,j} p_{i,j} \end{aligned} \tag{4}$$

where $U(\mathsf{c})$ 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 $\le x$. Note that the encoded packet of frame group Θ_X, i.e., $q_n(X)$, in SNC is the same as q_n in UNC. Similarly, if the stream selected for transmission is different from the stream that peer n watches, i.e., $s \ne \mathcal{S}(n)$, the generated NC packet is

$$q_n(x) = \sum_{q_m \in \mathcal{Q}_n} U(\Phi(q_m) \le x)\, b_m q_m = \sum_{p_{i,j} \in \Theta_x} c_{i,j} p_{i,j}. \tag{5}$$

A peer n_i can recover all $\sum_{F_i \in \Theta_x} B_i$ packets in frame group Θ_x of stream s once it has received $\sum_{F_i \in \Theta_x} B_i$ innovative packets of SNC types $\le x$. Fig. 3 shows a possible frame group assignment for a GOP of 15 frames with three frame groups. The probability of decoding F_1 is much higher than the other frames in frame groups 2 and 3. Since generally first I-frame F_1 of a GOP is the most important, by recovering only F_1, a large distortion can already be reduced.

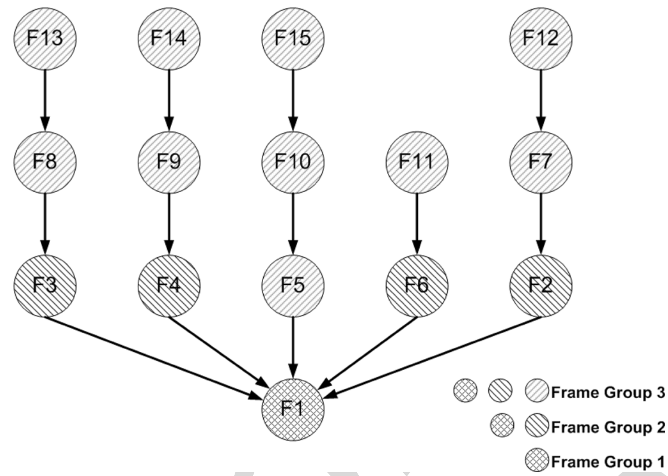

Fig. 3. DAG example with three frame groups.

IV. Packet Innovativeness

In this section, we estimate the innovative probability in a computation-efficient way. We first show a lower bound for the innovative probability for single stream case. Then by observing the differences between single and multi-stream, we estimate the innovative probability for the multi-stream case.

A. Innovative Probability for Single Stream

The exact computation of the NC packet innovative probability involves careful tracking of states of all peers in the CPR network. For example, [19] provided a complex innovative probability analysis for a gossip-based protocol, in which each peer in the network randomly selects another peer to send or to receive packets. Our CPR scenario is even more difficult in that each peer's transmission has multiple potential receivers because local WLAN broadcast is used. So instead of looking for an exact solution, we provide a simple and effective way of estimating the probability.

Suppose n_i transmits an NC packet to n_j using UNC. We denote B as the total number of packets needed to be disseminated for packet recovery; in the case of UNC, $B = P$. We also call B the *batch size*. We denote sets $\mathcal{S}_i = \{\mathbf{v}_i^1, \mathbf{v}_i^2, \ldots, \mathbf{v}_i^{k_i}\}$ and $\mathcal{S}_j = \{\mathbf{v}_j^1, \mathbf{v}_j^2, \ldots, \mathbf{v}_j^{k_j}\}$ as the native coefficient vectors of innovative native or NC packets in n_i and n_j before the transmission, respectively. Denote $\mathcal{SP}_i$ and $\mathcal{SP}_j$ as the subspaces spanned by the vectors in $\mathcal{S}_i$ and $\mathcal{S}_j$, respectively. Since vectors in $\mathcal{S}_i$ ($\mathcal{S}_j$) are linearly independent, they form a basis for subspace $\mathcal{SP}_i$ ($\mathcal{SP}_j$) with k_i (k_j) being the dimension of the subspace. n_j receiving an innovative packet means the coefficient vector associated with the received packet, together with vectors in $\mathcal{S}_j$, remain linearly independent. That means the innovative probability is also the probability that the dimension of $\mathcal{SP}_j$ increases.

We assume that the components in all native coefficient vectors take on values randomly chosen from $GF(O)$. This assumption is reasonable when peers are watching the same stream because in the UNC scheme all of the packets are treated equally and the encoding coefficients are also randomly chosen from $GF(O)$. We note that the assumption is less accurate at the beginning of the repairing process when the peers only have the chance to mix packets with neighbors close by. However it becomes more and more accurate with increasingly more packet mixing with peers. Let us define $P_{inv}^{k_i,k_j,B}$ as the *instantaneous innovative probability* of the received packet at peer n_j. We can summarize the lower bound for $P_{inv}^{k_i,k_j,B}$ with the following theorem.

Theorem 1: Assuming the dimensions of the subspaces spanned by the native coefficient vectors in peers n_i and n_j are k_i and k_j, then the instantaneous innovative probability of the NC packet transmitted from n_i to n_j has a lower bound as follows:

$$P_{inv}^{k_i,k_j,B} \geq \begin{cases} 1-\frac{1}{O}, & k_i > k_j \\ \left(1-\frac{1}{O}\right)(1-Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\}), & k_i \leq k_j \end{cases} \tag{6}$$

where $Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\}$ is the probability that the subspace spanned by vectors in $\mathcal{S}_i$ is a subset of the subspace spanned by the vectors in $\mathcal{S}_j$, which can be calculated as

$$Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\} = \frac{\prod_{t=0}^{k_i-1}(O^{k_j}-O^t)}{\prod_{t=0}^{k_i-1}(O^B-O^t)}. \tag{7}$$

Proof: We leverage [19, Lemma 2.1], which stated if the subspace spanned by native coefficient vectors in the transmitting peer is not a subset of the subspace spanned by the native coefficient vectors in the receiving peer, then the probability that the subspace dimension increases at the receiving peer, i.e., the innovative probability, is at least $1-1/O$. If dimension k_i of $\mathcal{SP}_i$ is larger than dimension k_j of $\mathcal{SP}_j$, then obviously $\mathcal{SP}_i \nsubseteq \mathcal{SP}_j$, and the first line of (6) follows.

The second line of (6) follows similar argument, and the key is to find $Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\}$ when $k_i \leq k_j$. Since $\mathcal{S}_i$ is a set of basis vectors for $\mathcal{SP}_i$, $Pr\{\mathcal{SP}_i \subseteq \mathcal{SP}_j\}$ is the same as the probability that each basis vector in $\mathcal{S}_i$ is also in $\mathcal{SP}_j$, i.e., $Pr\{\mathbf{v}_i^k \in \mathcal{SP}_j, \forall \mathbf{v}_i^k \in \mathcal{S}_i\}$. Since there are a total of O^B vectors over $GF(O)$, the first vector selected from $\mathcal{S}_i$ has O^B-1 possible choices excluding the zero vector. With k_j linearly independent vectors, there are O^{k_j} different vectors in subspace $\mathcal{SP}_j$. Then the probability that the first vector in $\mathcal{S}_i$ is in $\mathcal{SP}_j$ is $(O^{k_j}-1)/(O^B-1)$ where the "-1" in the numerator and denominator accounts for the zero vector. Similarly, the probability that the second vector in $\mathcal{S}_i$ is also in $\mathcal{SP}_j$ is $(O^{k_j}-O)/(O^B-O)$ where the "$-O$"accounts for vectors that are linear combinations of the first vector. Continue calculating the probabilities for the rest of the vectors in $\mathcal{S}_i$ and multiply all of them, we get the result for the second case. Combining the two cases, we have (6). □

Since our derivations are exact and the bound provided in [19, Lemma 2.1] is achievable, the result in Theorem 1 is tight and is achievable. Equation (6) shows the innovative probability assuming dimensions of the subspaces $\mathcal{SP}_i$ and $\mathcal{SP}_j$ are known. Generally, we define the *probability mass function* (PMF) of the dimensions of the subspaces for the *average peer* n as $f(k)$, and we can calculate the lower bound of the *average innovative probability*, P_{inv}^B, by a weighted average

$$P_{inv}^B = \sum_{k_i=1}^{B}\sum_{k_j=1}^{B} P_{inv}^{k_i,k_j,B} f(k_i) f(k_j). \tag{8}$$

B. Innovative Probability for Multi-Stream

When there are multiple streams being repaired simultaneously, our assumption that the components of the native coefficient vectors are randomly generated from $GF(O)$ is altered. This is because when a peer forwards a stream that he/she is not watching, he/she can only encode a packet using packets received from other peers through CPR without any packets received directly from WWAN. Without the chance of mixing the packets, the randomness of the components in the native coefficient vectors is reduced and thus our previous assumption does not hold.

To better understand the problem, let us consider a scenario where all peers are repairing two streams: s^1 and s^2. Assuming peers randomly select one stream to watch, then for a peer n watching stream s^1, half of n's neighbors are also watching s^1, and they can each send NC packets to n with innovative probability P_{inv}^B. The innovative probability of NC packets sent from the other half of n's neighbors to n, who are watching stream s^2, depends in turn on their neighbors, i.e., two-hop neighbors of n. Again, with probability 1/2, n's two-hop neighbors are watching s^1 and can help n via n's one-hop neighbors. For the rest half two-hop neighbors that watch s^2 can also receive some packets of stream s^1 during the repairing process, and with these limited packets they can help as well.

At this point, we need to consider the *common neighbor* effect where n's one-hop neighbors can receive identical packets from the same two-hop neighbor of n. Note we do not apply this effect to the two-hop neighbors who watch s^1 because different common one-hop neighbors may belong to many common neighbor groups and they can receive different packets from those two-hop neighbors during the CPR process, which greatly reduces the effect. However, this is not true for the two-hop neighbors who watch stream s^2 and have limited packets belonging to s^1.

The common neighbor effect is illustrated in Fig. 4, where peers n_5 and n_6 receive the same packet of stream s^1 from n_2, which reduces the innovative probability of subsequent NC packets forwarded to n by half. The innovative probability for the two stream scenario can now be estimated as $(1/2)P_{inv}^B + (1/2)\left(P_{inv}^B/4 + P_{inv}^B/2\right)$.

In general, denoting the average number of common neighbors as N_c, the average innovative probability for multi-stream is estimated as

$$\begin{aligned} P_{inv}^{B,S} &\approx \frac{1}{S}P_{inv}^B + \left(1-\frac{1}{S}\right)\left[\frac{P_{inv}^B}{N_c-\frac{N_c}{S}}\left(1-\frac{1}{S}\right) + P_{inv}^B\frac{1}{S}\right] \\ &= \left(2-\frac{1}{S}\right)\frac{P_{inv}^B}{S} + \left(1-\frac{1}{S}\right)\frac{P_{inv}^B}{N_c}. \end{aligned} \tag{9}$$

The first term in (9) accounts for neighbors watching the same stream as the receiving peer under consideration, and the second term accounts for neighbors watching different streams. Note that our derivation is limited to two-hop neighbors, which is conservative.

When SNC is considered, the innovative probability is estimated similarly as in the UNC case, except we set the batch size B to the size of the frame group that is under repair. Note that although we can get the simulated innovative probability under some scenarios offline, we cannot get it under all cases because in practice the topology of the network may change and N_c may change. In the following, we will use the analytical innovative probability for SNC optimization.

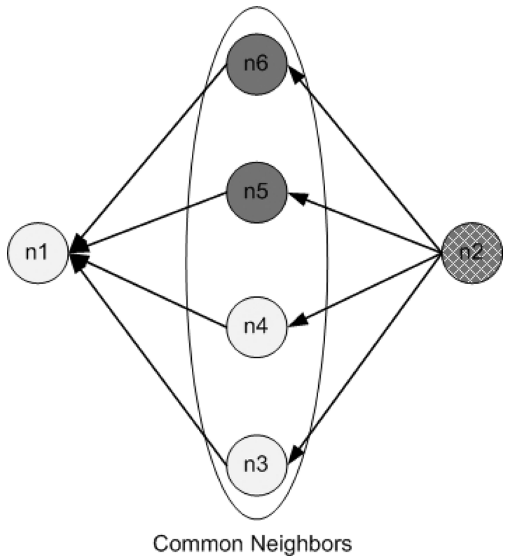

Fig. 4. Common neighbors in CPR network. n_3, n_4, n_5, and n_6 are common neighbors of n_1 and n_2. n_1, n_3, and n_4 watches s^1, and n_2, n_5, and n_6 watches s^2. n_2 receives one packet of s^1 during the repair process.

V. SNC Optimization Framework

In this section, we propose a framework to optimize structures and transmissions of network-coded CPR packets at peers so that the expected distortions of streams are minimized. Our proposed SNC optimization has two steps. First, the media source defines a global NC structure to minimize distortion for the average peer with average connectivity. Second, at each transmission opportunity a peer selects a stream from $\mathcal{A}_n$ and a type within the defined NC structure to transmit given its available local state information of its neighbors. We discuss the two steps in order.

A. Global NC Structure Definition

The media source first optimizes an NC structure for each stream s for the *average peer n*, assuming that an average peer can expect R_n packets from neighbors during CPR. Using the DAG source model from Section II-B, the expected distortion at peer n watching stream s can be written as

$$\Delta_n = D - \sum_{i=1}^{M} d_i \prod_{j \preceq i} \alpha_n(j). \tag{10}$$

d_i is calculated as the additional PSNR improvement of using decoded frame i for display of frame i, plus the PSNR improvement of using decoded frame i for error concealment of descendant frames of frame i in the source dependency tree in the event that they are incorrectly decoded, minus the PSNR improvement of using the parent frame of frame i (if one exists) for error concealment of frame i and its descendant frames. D is the sum of all d_i in one GOP, i.e., the distortion when no frame is received. $\alpha_n(j)$ is the recovery success probability of frame F_j at peer n. Note that in (10) we make the simplifying assumption that the frame recovery probability is independent from each other.

$\alpha_n(j)$ itself can be written as

$$\alpha_n(j) = (1-l)^{B_j} + \left(1-(1-l)^{B_j}\right) S_n(j) \tag{11}$$

where l is the WWAN 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 WWAN.

Suppose we are given SNC groups $\Theta_1, \ldots, \Theta_X$. 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. We can hence write $S_n(j)$ as

$$S_n(j) \approx Q(n, g(j)) + \sum_{y=g(j)+1}^{X} Q(n, y) \prod_{z=g(j)+1}^{y} (1 - Q(n, z-1)) \tag{12}$$

where $Q(n, x)$ is the probability that peer n can NC-decode SNC type x by receiving $\sum_{F_i \in \Theta_x} B_i$ innovative native or NC packets. Note here we make the simplifying assumption that the recoveries of the frame groups are uncorrelated.

Using the average innovative probability shown in (8), if a peer n sends a NC packet of type x with probability $\beta_n(x)$, we can approximate $Q(n, x)$ as in (13) at the bottom of the page, where $L_j = \sum_{F_i \in \Theta_j} B_i$ is the number of packets in group j. $\Omega_x = \lceil l L_x \rceil$ is the expected number of lost packets of type x due to WWAN broadcast and needed CPR repairs. $1/S$ is the probability of receiving a particular stream given an active set of S streams. In words, (13) finds the frame group recovery probability by looking at the complimentary event that the frame group cannot be recovered, i.e., less than Ω_x innovative packets of SNC types $\leq x$ are received. Among the expected received R_n CPR packets, k of them are of SNC types $\leq x$ and are innovative. These packets are useful for n to recover frame group x. For the rest $R_n - k$ packets, some of them are of SNC types $\leq x$

$$Q(n, x) \approx 1 - \sum_{k=0}^{\Omega_x - 1} \binom{R_n}{k} \left(\frac{1}{S} \sum_{i=1}^{x} \beta_n(i) P_{inv}^{L_i, S} \right)^k \sum_{t=0}^{R_n - k} \binom{R_n - k}{t} \times \left(\frac{1}{S} \sum_{i=x+1}^{X} \beta_n(i) + \frac{S-1}{S} \right)^t \left(\frac{1}{S} \sum_{i=1}^{x} \beta_n(i) \left(1 - P_{inv}^{L_i, S}\right) \right)^{R_n - k - t} \tag{13}$$

but are not innovative; some of them are of SNC types greater than x. These packets are not useful for n to recover frame group x.

With our formulation shown in (10) —(13), the SNC optimization at the media source is 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 as follows:

$$\min_{X,\{\Theta_x\},\{\beta_n(x)\}} \Delta_n. \tag{14}$$

To solve the optimization problem in (14), a simple exhaustive search scheme has been shown to be of exponential complexity [12]. We therefore used an efficient local search algorithm for fast optimization.

We first notice that the search space can be reduced by considering the DAG structure described in Section II-B. 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 NC types then, we will assign preceding frames with a smaller or equal NC type than succeeding frames given the DAG structure.

Based on the reduced search space, we perform the local search as follows. We first assign M NC 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 NC type smaller than F_i. For this NC structure, we exhaustively search the best $\beta_n(x)$ resulting in the smallest distortion using (14). We then find the best "merging" of parent and child frames—assigning the same NC 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.

With our local search scheme, we need to check at most M merging operations for M frames in each iteration, and there are at most M iterations. Hence there are at most M^2 merge operations performed, which is significantly less than the exhaustive search. In practice, M is small, and by restricting the search space of $\beta_n(x)$ to 0.1—0.9 with 0.1 increment, we can bound the optimization in a reasonable amount of time, which facilitates real-time video streaming.

B. SNC Local Peer Optimization

1) Peers Utilize Local State Information: In the previous section, an NC structure was globally optimized for the entire ad-hoc network assuming an average peer with average connectivity. During CPR, however, *local state information* can be easily exchanged among neighbors by piggybacking on data packets with minimal overhead. By local we mean only one-hop neighbor information. Specifically, we assume each NC packet from peer n reveals which stream the packet is repairing and which stream n is watching [$\mathcal{S}(n)$]. The NC packet also includes two state reports: 1) *native packet reception report* identifying which packets of stream $\mathcal{S}(n)$ were successfully delivered from WWAN, and 2) *NC group status report* containing the number of innovative packets that are received in each NC groups of $\mathcal{S}(n)$. Note that the obtained local neighbor information can become inaccurate (stale) over time.

Using local information, a peer first selects a stream among $\mathcal{A}_n$ for repair deterministically instead of picking one at random. For a chosen stream, a peer then selects a NC packet type to transmit deterministically. This can potentially further improve streaming performance locally beyond the global optimization performed in previous section; for example, if a peer's neighbors have already fully recovered a certain stream, then the peer will not choose that stream for repair.

2) Local Peer Optimization: Using the local information discussed above, at each transmission opportunity a peer can select the optimal stream for repair and the SNC type that results in the minimum total distortion among all its neighbors. More specifically, we optimize the following expression:

$$\min_{v\in\mathcal{A}_n,u\in\mathcal{U}_n^v} \sum_{m\in\{n\text{'s neighbors}\}} \Delta_m^{v,u} \tag{15}$$

where v and u are the stream and the SNC type to be decided for packet transmission. $\mathcal{U}_n^v$ is the set of SNC types in stream v peer n has. Similar to (10), $\Delta_m^{v,u}$, the resulting distortion of neighbor m when NC packet of type u in stream v is transmitted, is written as

$$\Delta_m^{v,u} = D - \sum_{i=1}^{M} d_i \prod_{j\preceq i} \alpha_m^{v,u}(j). \tag{16}$$

Note here the distortion reduction is for neighbor m, and D, M, and d_i are constants for stream $\mathcal{S}(m)$. Since peer n has local information from neighbor m, we have

$$\alpha_m^{v,u}(j) = \begin{cases} 1, & \text{if frame } j \text{ has been received} \\ S_m^{v,u}(j), & \text{otherwise.} \end{cases} \tag{17}$$

Note that the first line in (17) has two meanings: either all the packets in frame j of stream $\mathcal{S}(m)$ are successfully delivered through WWAN or they have been repaired through CPR. They are inferred from the native packet reception report and the NC group status report, respectively. $S_m^{v,u}(j)$ has similar formulation as in the global NC definition part except here we need to decide the stream and packet type for transmission. It is now approximated as

$$Q^{v,u}(m,g(j)) + \sum_{y=g(j)+1}^{X} Q^{v,u}(m,y) \prod_{z=g(j)+1}^{y} \left(1 - Q^{v,u}(m,z-1)\right). \tag{18}$$

Since peers now have neighbor information, $Q^{v,u}(m,x)$ is updated as in (19) at the bottom of the next page, where $L_m^{x,v,u}$

is the number of innovative packets of type $\leq x$ peer m needs to recover frame group x, which can be written as

$$\begin{cases} C_m^x - R_m \frac{t}{Y}\frac{1}{S}\sum_{i=1}^{x}\beta_m(i)P_{inv}^{L_i,S} - P_{inv}^{L_x,S}, & v=\mathcal{S}(m), u=x \\ C_m^x - R_m \frac{t}{Y}\frac{1}{S}\sum_{i=1}^{x}\beta_m(i)P_{inv}^{L_i,S}, & \text{otherwise.} \end{cases} \tag{20}$$

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. $R_m(t/Y)(1/S)\sum_{i=1}^{x}\beta_m(i)P_{inv}^{L_i,S}$ represents the estimated number of innovative packets of type $\leq x$ in stream $\mathcal{S}(m)$ neighbor m could receive during time interval t. If the transmitted stream v is the same as the stream peer m needs, $\mathcal{S}(m)$, and the transmitted packet type u is the same as x, then the packet transmitted from n to m will be an innovative packet with probability $P_{inv}^{L_x,S}$, which results in a reduction in the needed number of packets. Similarly, U_m is the total number of packets neighbor m could possibly receive during the rest of the repair time. It is written as

$$U_m = \left\lfloor R_m\left(1 - \frac{t'}{Y}\right)\right\rfloor - 1 \tag{21}$$

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

Note that in (19) and (20), we assume conservatively that peer m's other neighbors do not perform local optimization, but instead are transmitting using the predetermined transmission probability. This is due to the fact that to predict the optimization results of peer 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 difficult to achieve in a distributed scenario.

VI. Simulation Studies

In this section, we verify the effectiveness of our SNC optimization framework through simulations. We first present the simulation setup: the video codec parameters and the CPR network settings. Next, we show the result of the innovative probability estimation. We then compare the performance of the UNC and SNC schemes when CPR bandwidth is not sufficient to repair all WWAN losses for each stream. Finally, we examine the benefits of the two proposed innovations in our SNC framework: local peer optimization and innovative probability estimation.

A. Simulation Setup

Two test video sequences were used for simulations: 300-frame MPEG class A `news` and class B `foreman` sequences at QCIF resolution (176×144), at 30 fps and sub-sampled in time by 2. The GOP size was chosen at 15 frames: one I-frame followed by 14 P-frames. Quantization parameters used for I-frames and P-frames were 30 and 25, respectively. The H.264 codec used was JM 12.4, downloadable from [20]. We performed reference frame selection in [15] with target encoding rate at 220 kbps, resulting in a DAG describing inter-frame dependencies as discussed in Section II-B. For each trial, we used the same video sequence as media content for all streams. A peer selected a stream to watch randomly among all available streams.

We considered a CPR network of size $1000 \times 1000\ \mathrm{m}^2$ where 50 peers were uniformly distributed. The peers were watching video streams through MBMS using their multi-homed devices, where WLAN interfaces were activated for CPR. We used the broadcast mode of WLAN, therefore no feedback messages were sent from the receivers and no transmission rate adaption was performed. The media source provided S_{all} streams, each of which was transmitted at rate $r_{\text{MBMS}} = 220$ kbps. Given one GOP was 15 frames and video was encoded at 15 fps, one epoch time Y is 1 s. The MBMS broadcast packet loss rate was kept constant at 0.1. Each CPR packet is set to the size $W = 1000$ bytes. We used QualNet [21] to conduct the simulations. To have the freedom to vary CPR bandwidth, we selected *Abstract PHY* in QualNet for physical layer and set all of the parameters to be the default values in 802.11.

B. Simulation Results

1) Innovative Probability: We compared our analytical results on innovative probability to the simulation results in this section. Simulations for both the single stream and multi-stream scenarios were performed. The video sequence in use was the `news` sequence. The CPR bandwidth was 4.5 Mbps, which is the typical data rate for 802.11b.

Fig. 5(a) plots the average innovative probability when all the peers were watching the same stream and used UNC scheme to do the repairing. Since the average number of initial packet loss was lB, where l is MBMS packet loss rate, we assumed that PMF $f(k)$ was uniformly distributed between $(1-l)B$ and B. This assumption is reasonable because during the repairing process, the dimensions of the encoding coefficient vectors were increasing gradually and steadily. Because of the

$$Q^{v,u}(m,x) \approx 1 - \sum_{k=0}^{\lceil L_m^{x,v,u}\rceil - 1}\binom{U_m}{k}\left(\frac{1}{S}\sum_{i=1}^{x}\beta_m(i)P_{inv}^{L_i,S}\right)^k \sum_{t=0}^{U_m-k}\binom{U_m-k}{t} \times\left(\frac{1}{S}\sum_{i=x+1}^{X_m}\beta_m(i) + \frac{S-1}{S}\right)^t\left(\frac{1}{S}\sum_{i=1}^{x}\beta_m(i)\left(1-P_{inv}^{L_i,S}\right)\right)^{U_m-k-t} \tag{19}$$

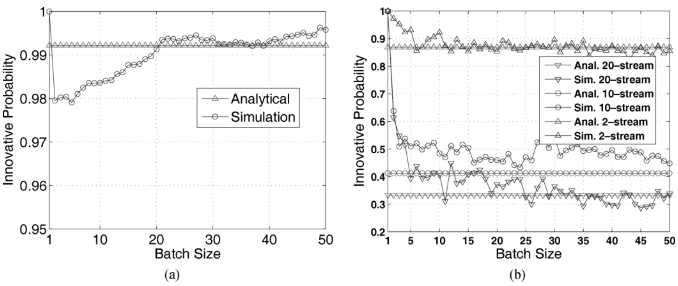

Fig. 5. Receiving CPR packet innovative probability. a) Single stream. b) Multi-stream.

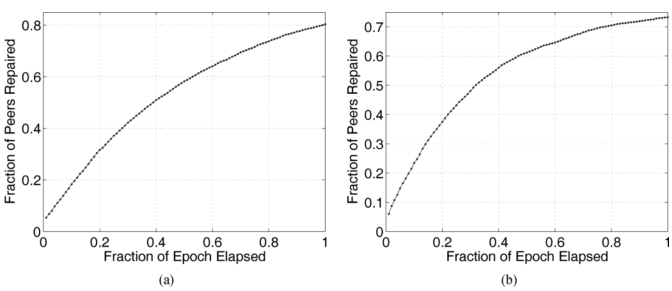

Fig. 6. CDF of the number of peers repaired during one epoch time. a) CPR BW 4.5 Mbps, $S = 10$. b) CPR BW 23 Mbps, $S = 20$.

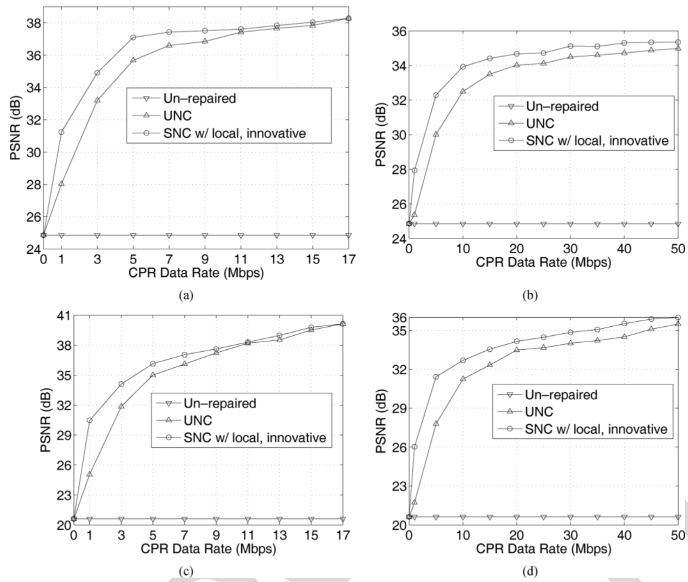

Fig. 7. PSNR for `news` and `foreman` under various CPR data rates. a) `news` ten streams. b) `news` 20 streams. c) `foreman` ten streams. d) `foreman` 20 streams.

low packet loss rate, peers received most of the packets from MBMS. Therefore each transmitted NC packet is a combination of a large number of native and NC packets, which makes the components of the native coefficient vectors random and the innovative probability close to 1. The difference between the analytical and simulation results was small and was due to the simplified assumption of uniform distribution on the dimension of subspaces.

Fig. 5(b) shows the analytical result versus the simulation result under various multi-stream scenarios. Intuitively, with the increase of the number of video streams, the innovative probability is reduced. We see that the analytical results capture the trend of the simulation results very well.

2) Multi-Stream Repair With UNC: As discussed in Section III, if a peer does not receive a sufficient number of innovative native or NC packets during CPR to recover *all* WWAN losses, then UNC could not recover *any* lost packets using received NC packets. This undesired phenomenon was depicted in Fig. 6(a), which shows the CDF of the fraction of peers that recovered all packets through CPR in one epoch time using UNC. There were $S = 10$ total active streams, and on average 5 peers were watching the same stream. CPR operated at the typical 802.11b data rate. As shown, only about 80% of peers recovered their lost packets in one epoch time. Similarly, Fig. 6(b) shows the CDF when there were $S = 20$ total active streams, and the CPR bandwidth was increased to 23 Mbps, the typical data rate for 802.11a/g. The result was similar, and fewer than 75% of the peers benefited from CPR with UNC.

3) Multi-Stream Repair With SNC: We now show the performance of SNC for the multi-stream scenario. The complete SNC scheme involves a two-step optimization: 1) media source first searches for the optimal NC structure for each stream separately using the optimization framework shown in Section V; and 2) individual peer performs local optimization by utilizing partial state information received from neighbors. When a peer has received enough packets for a certain frame group, the packets within that particular frame group can be recovered. With our SNC frame group optimization, it turned out that when the CPR bandwidth was low, the SNC optimization returned more NC types than when the bandwidth was high. We also noted that the lower the bandwidth was, the smaller the sizes of the first few NC groups. This is reasonable because when bandwidth is low, peers need desperately to decode at least the first few frames. Dividing the packets into more groups increases the chance that the received packets can be decoded, and therefore peers can at least decrease some of the distortion with the limited number of receiving packets.

In the following, we first compare the performance of SNC to UNC under different CPR data rates using different video sequences. We then show the effectiveness of the local peer optimization and the innovative probability estimation in the SNC optimization framework. Lastly we explore how the number of streams affected the performance.

SNC Outperforms UNC: Fig. 7(a) and (b) shows the CPR data rates versus PSNR plot for `news` when there were ten and 20 streams, respectively. Fig. 7(c) and (d) shows the CPR data rates versus PSNR plot for `foreman`. We also have the un-repaired video quality, the original video quality without any CPR repairs, as a performance benchmark.

From Fig. 7 it can be easily observed that SNC outperformed traditional UNC and un-repaired video in all transmission rates. When there were ten streams provided by MBMS, SNC provided up to 13.51 dB PSNR improvement for the `news` sequence and 19.71 dB PSNR improvement for the `foreman` sequence over un-repaired video when the data rate was larger than 17 Mbps. When there were 20 streams, the performance improvement over un-repaired video using SNC were up to 10.51 dB and 15.37 dB when the data rate was larger than 50 Mbps. For UNC, the peers needed $\sum_{j=1}^{15} B_j$ innovative native or NC packets before any repairing could be performed. However, for the SNC scheme, peers could repair important frames as soon as sufficient NC packets of particular SNC types were received. Hence when bandwidth was low, the performance of SNC was much better than UNC. For example, at the transmission rate of 1 Mbps, SNC achieved 3.21 dB gain over UNC for the `news` sequence and around 5.39 dB gain for the `foreman` sequence where there were ten streams. When the bandwidth was higher, the number of received packets increased so that UNC recovered more packets and the performance of the two schemes became similar. Note that when there were ten streams, when the 802.11 data rate exceeded 17 Mbps, all the packets could be repaired for both `news` and `foreman`. However when there were 20 streams, even when the 802.11 data rate was almost at maximum, 50 Mbps, there were still packet loss. Therefore it is always better to choose SNC over UNC when the number of streams is large. We note that with the increase of CPR data rate, the slopes of the curves were reducing. We explain this phenomenon with following three reasons: 1) with the increase of CPR data rate, the packet loss rate was also increased, which reduced the effective bandwidth; 2) distortions of the frames in a GOP was not uniformly distributed. With the first few received packets, more distortion could be recovered through CPR; 3) the packet innovative probability reduced with the increased number of receiving packets.

Comparing the video qualities for the `news` and `foreman` sequences, we found that the improvement by using SNC over the UNC scheme was more pronounced for the `foreman` sequence. For example, as shown earlier the gain was 3.21 dB for the `news` sequence and 5.39 dB for the `foreman` sequence when ten streams were repaired under 1 Mbps CPR data rate. This is due to the fact that `foreman` has more inherent motion and requires more encoding bits for the same given quantization parameters. As a result, the corresponding DAG was long rather than wide, which means that if a particular packet close to the root node is lost, it affects many descendant frames and results in large distortion.

Effectiveness of Local Peer Optimization and Innovative Probability Estimation: We also examine the individual benefits of the two innovations we propose within the SNC framework: local peer optimization and innovative probability estimation. We compare the performance when: 1) both innovations were removed; 2) only innovative probability estimation was added; and 3) both innovations were added.

Fig. 8(a) and (b) compares the performance of SNC under different configurations for both the `news` and `foreman` sequences. First, note that SNC without both innovations already outperformed UNC for all configurations. For example at 1 Mbps CPR data rate, for the `news` sequence and without local optimization and innovative probability estimation (innovative probability set to 1), SNC achieved a gain of 1.54 dB over UNC. When we used innovative probability estimation only, we reaped 2.65 dB gain over the UNC scheme. By utilizing both local peer optimization and innovative probability estimation, SNC provided 3.21 dB gain over UNC. The results were similar for the `foreman` sequence.

Number of Streams Affects Performance: Fig. 9 shows the performance of UNC and SNC when the stream number varied

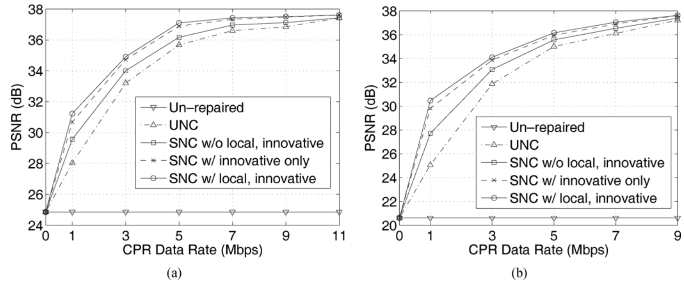

Fig. 8. PSNR for the `news` and `foreman` sequences under various CPR transmission rates and SNC scheme settings. a) `news` ten streams. b) `foreman` ten streams.

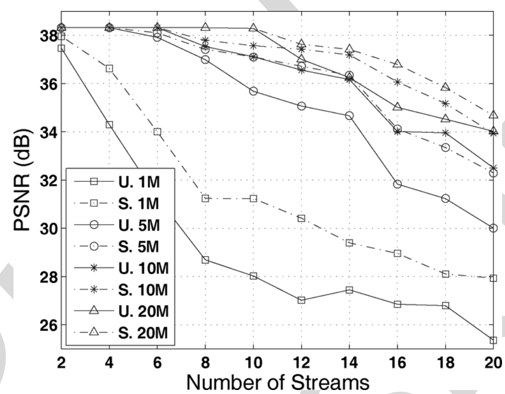

Fig. 9. PSNR for the `news` sequence under various multi-stream scenarios. U. and S. are short for UNC and SNC, respectively.

from 2 to 20. Obviously with the increase of the number of video streams, performance decreased because the CPR bandwidth that could be allocated to a particular stream was reduced. Peers had to contribute most of their CPR bandwidth to help others. Nevertheless, our SNC scheme showed noticeable gain over the UNC scheme for all cases.

VII. Related Work

Due to the aforementioned NAK implosion problem [3], many video streaming strategies over MBMS [4] have forgone feedback-based error recovery schemes like [2] and opted instead for forward error correction (FEC) schemes like Raptor codes [4]. 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. We instead exploit the multi-homed nature and propose to repair lost packets through CPR.

NC has been a popular research area since Ahlswede's seminal work [22], which showed that network capacity can generally be achieved using NC. Many studies have since explored message dissemination using NC. In [23], the authors proposed to use random NC [17] to encode the packets to be transmitted in a peer-to-peer content delivery scenario. We leverage this idea to our design and focus on video streaming and NC structure in wireless ad-hoc networks. A gossip-based protocol was proposed in [19] which utilizes network coding to disseminate messages. Instead of gossiping, we utilize the broadcast nature of the wireless medium to disseminate video packets.

Recent works [18], [24]–[26] have attempted to jointly optimize video streaming and NC. [18] 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 GOP, while [18] is performed greedily per packet.

Reference [24] utilized the hierarchical NC scheme in the same way for CDN and P2P networks to combat Internet bandwidth fluctuation. Our work is more general in that our source model is a DAG, while the model in [24] is a more restricted dependency chain. Moreover, we provide a NC optimization framework to better exploit the benefit of SNC.

[25] discussed the application of Markov Decision Process [16] to NC, in which NC optimization and scheduling are centralized at the access point or base station. Like [18] 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 work, we instead consider fully distributed peer-to-peer repair without assuming full knowledge of state information of peers.

Reference [26] discussed applying structure on NC across multiple generations of video packets, where one generation is defined at the transport layer irrespective of application-layer GOP structures. In our work, NC is applied within one GOP, and the structure is defined according to the dependency tree among the video frames in the GOP. Defining NC structure within a GOP enables us to build a rate-distortion based NC optimization framework which finds the optimal NC structure resulting in the smallest expected distortion. To our knowledge, we are also the first in the NC literature to use randomization in the implementation of SNC for video streaming optimization.

VIII. CONCLUSIONS

In this paper, we present a novel, rate-distortion optimized, NC-based, cooperative peer-to-peer packet repair solution for the multi-stream WWAN video broadcast. We make contributions in the following major aspects. First, we propose a two-step NC structure optimization framework in which the video stream repair can be optimized in a rate-distortion manner. Second, we analyze the innovative probability of a receiving NC packet to facilitate accurate NC structure optimization. Lastly, we provide detailed simulations and show that the video quality can be improved by up to 19.71 dB over un-repaired video stream and by up to 5.39 dB over video stream using traditional unstructured network coding.

REFERENCES

[1] 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, 2006.

[2] G. Cheung, W.-T. Tan, and T. Yoshimura, "Double feedback streaming agent for real-time delivery of media over 3G wireless networks," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 304–314, Apr. 2004.

[3] J. Crowcroft and K. Paliwoda, "A multicast transport protocol," in *Proc. ACM SIGCOMM*, New York, Aug. 1988.

[4] J. Zfzal, T. Stockhammer, T. Gasiba, and W. Xu, "Video streaming over MBMS: A system design approach," *J. Multimedia*, vol. 1, no. 5, pp. 25–35, Aug. 2006.

[5] P. Sharma, S.-J. Lee, J. Brassil, and K. Shin, "Distributed communication paradigm for wireless community networks," in *Proc. IEEE Int. Conf. Communications*, Seoul, Korea, May 2005.

[6] S. Raza, D. Li, C.-N. Chuah, and G. Cheung, "Cooperative peer-to-peer repair for wireless multimedia broadcast," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, Beijing, China, Jul. 2007, pp. 1075–1078.

[7] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, "Network coding-based broadcast in mobile ad-hoc networks," in *Proc. IEEE INFOCOM 2007*, Anchorage, AL, May 2007, pp. 1739–1747.

[8] X. Liu, S. Raza, C.-N. Chuah, and G. Cheung, "Network coding based cooperative peer-to-peer repair in wireless ad-hoc networks," in *Proc. IEEE Int. Conf. Communications (ICC)*, Beijing, China, May 2008.

[9] A. Rahmati and L. Zhong, "Context-for-wireless: Context-sensitive energy-efficient wireless data transfer," in *Proc. ACM MobiSys*, San Juan, PR, Jun. 2007, pp. 165–178.

[10] S. Chandra and A. Vahdat, "Application-specific network management for energy-aware streaming of popular multimedia formats," in *Proc. USENIX Annu. Conf.*, Monterey, CA, Jun. 2002, pp. 329–342.

[11] T. Pering, Y. Agarwal, R. Gupta, and R. Want, "Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces," in *Proc. ACM MobiSys*, Jun. 2006, pp. 220–232.

[12] 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 *Proc. 1st IEEE Workshop on Mobile Video Delivery*, Newport Beach, CA, Jun. 2008.

[13] X. Liu, G. Cheung, and C.-N. Chuah, "Structured network coding and cooperative local peer-to-peer repair for MBMS video streaming," in *Proc. Int. Workshop on Multimedia Signal Processing*, Cairns, Queensland, Australia, Oct. 2008.

[14] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[15] G. Cheung, W.-T. Tan, and C. Chan, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 649–662, Jun. 2007.

[16] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[17] 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 Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[18] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," in *Proc. IEEE 16th Int. Packet Video Workshop*, Lausanne, Switzerland, Nov. 2007, pp. 191–200.

[19] S. Deb, M. Medard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2486–2507, Jun. 2006.

[20] The TML Project Web-Page and Archive. [Online]. Available: http://iphome.hhi.de/suehring/tml/.

[21] Qualnet. [Online]. Available: http://www.scalable-networks.com/.

[22] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeun, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[23] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE Infocom*, Mar. 2005, pp. 2235–2245.

[24] K. Nguyen, T. Nguyen, and S.-C. Cheung, "Video streaming with network coding," *Springer J. Signal Process. Syst Special Issue: ICME07*, Feb. 2008.

[25] D. Nguyen, T. Nguyen, and X. Yang, "Multimedia wireless transmission with network coding," in *Proc. IEEE 16th Int. Packet Video Workshop*, Lausanne, Switzerland, Nov. 2007, pp. 326–335.

[26] M. Halloush and H. Radha, “Network coding with multi-generation mixing: Analysis and applications for video communication,” in *Proc. IEEE Int. Conf. Communications*, May 2008.

Xin Liu (S’XX) **AUTHOR: MISSING MEMBER YEARS.** received the B.S. degree in electronic and information engineering and the M.E. degree in computer engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2003 and 2005, respectively. In 2006, he received the M.S. degree in electrical engineering from the State University of New York at Buffalo, Buffalo, NY. He is pursuing the Ph.D. degree in computer engineering at the University of California, Davis.

His research interests are wireless networking, cooperative communications, and network coding.

Gene Cheung (SM’XX) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1995 and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1998 and 2000, respectively.

In August 2000, he joined Hewlett-Packard Laboratories Japan, Tokyo, where he is currently a Senior Researcher. His research interests include media transport over wireless networks, joint source/network coding for video streaming, and community-based media interaction.

Dr. Cheung is currently an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and a voting member of the IEEE Multimedia Communications Technical Committee.

Chen-Nee Chuah (SM’XX) received the B.S. degree in electrical engineering from Rutgers University, New Brunswick, NJ, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley.

She is currently an Associate Professor in the Electrical and Computer Engineering Department at the University of California, Davis. Her research interests include Internet measurements, network management, and wireless/mobile computing.

Dr. Chuah received the NSF CAREER Award in 2003 and the Outstanding Junior Faculty Award from the UC Davis College of Engineering in 2004. In 2008, she was selected as a Chancellor’s Fellow of UC Davis. She has served on the executive/technical program committee of several ACM and IEEE conferences and is currently an Associate Editor for IEEE/ACM TRANSACTIONS ON NETWORKING.