Distributed Markov Decision Process in Cooperative Peer Recovery for WWAN Multiview Video Multicast

Zhi Liu [#*1], Gene Cheung [*2], Yusheng Ji [*3]
[#] The Graduate University for Advanced Studies, [*] National Institute of Informatics
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan 101-8430
Email: {[1] liuzhi, [2] cheung, [3] kei}@nii.ac.jp

Abstract— **Error resilient video multicast over Wireless Wide Area Networks (WWAN) is difficult because of unavoidable packet losses and impracticality of retransmission on a per packet, per client basis due to the well-known NAK implosion problem. In response, Cooperative Peer-to-peer Repair (CPR) calls for multi-homed devices listening to the same video multicast to locally exchange received WWAN packets via a secondary network like ad hoc Wireless Local Area Network (WLAN) to alleviate individual WWAN packet losses. When videos of the interested 3D scene are captured by multiple closely spaced cameras, each video can be encoded into a separate video stream and transmitted on its own WWAN multicast channel. Clients can then switch observation viewpoints periodically by simply re-subscribing to different WWAN multicast channels—a scenario called interactive multiview video streaming (IMVS). IMVS complicates the CPR WWAN loss recovery process, however, since neighbors of a loss-stricken peer can now be watching different views. In this paper, we optimize the decision process for individual peers during CPR for recovery of multiview video content in IMVS. In particular, for each available transmission opportunity, a peer decides—using Markov decision process as a mathematical formalism—whether to transmit, and if so, how the CPR packet should be encoded using structured network coding (SNC). A loss-stricken peer can then either recover using received CPR packets of the same view, or using packets of two adjacent views and subsequent view interpolation via image-based rendering. Experiments show that decisions made using our proposed MDP outperforms decisions made by a random scheme by at least $1.8dB$ in PSNR in received video quality in typical network scenario.**

I. Introduction

Video multicast over the Wireless Wide Area Network (WWAN) remains challenging due to packets' stringent playback deadlines and unavoidable packet losses in the wireless transmission medium. Moreover, due to the well-known NAK implosion problem [1], server cannot perform retransmission to recover packets lost during WWAN multicast on a per-packet, per-client basis. A conventional approach [2] is to use Forward Error Correction (FEC) packets to protect source packets. However, deploying sufficient amount of FEC even for the worst-channel peer in the multicast group translate to a large overhead, leaving preciously few bits out of a finite WWAN transmission budget for source coding to fight quantization noise, resulting in poor video quality.

One alternative solution to alleviate the wireless packet loss problem is *Cooperative Peer-to-peer Repair* (CPR) [3]. CPR, exploiting the "uncorrelatedness" of neighboring peers' channels to the same WWAN source (hence unlikely for all peers to suffer bad channel fades at the same time), calls for peers to locally exchange received WWAN packets via a secondary network such as an ad hoc Wireless Local Area Network (WLAN). Further, it was shown [3] that instead of exchanging raw received WWAN packets from the server, a peer can first encode a *structured network coding* (SNC) repair packet using received source packets before sharing the encoded repair packet to further improve packet recovery.

When videos of the interested 3D scene are captured by multiple closely spaced cameras, each video can be encoded into a separate video stream and transmitted on its own WWAN multicast channel. Clients can then switch observation viewpoints periodically by simply re-subscribing to different WWAN multicast channels—a scenario called interactive multiview video streaming (IMVS) [4]. IMVS complicates the CPR WWAN loss recovery process, however, since neighbors of a loss-stricken peer can now be watching different views.

In this paper, we optimize the decision process for individual peers during CPR for recovery of multiview video content in IMVS. In particular, for each available transmission opportunity, a peer decides—using Markov decision process as a mathematical formalism—whether to transmit, and if so, how the CPR packet should be encoded using structured network coding (SNC). A loss-stricken peer can then either recover using received CPR packets of the same view, or using packets of two adjacent views and subsequent view interpolation via image-based rendering (IBR) [5]. Our proposed MDP is fully distributed and peer-adaptive, so that state transition probabilities in the MDP can be appropriately estimated based on observed aggregate behavior of neighboring peers. Experiments show that decisions made using our proposed MDP outperforms decisions made by a random scheme by at least $1.8dB$ in PSNR in received video quality.

The outline of the paper is as follows. We first overview our streaming system in Section II. We then discuss the construction of MDP for CPR packet selection in Section III. We discuss experimentation in Section IV.

II. System Overview

A. WWAN Multiview Video Multicast with CPR

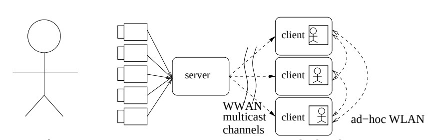


Fig. 1. Overview of WWAN Multiview Video Multicast System

The components of our proposed WWAN multiview video multicast system, shown in Fig. 1, are as follows. M cameras in a one-dimensional array capture a scene of interest from different viewpoints. A server compresses the M different views into M individual streams and transmits them, synchronized in time, in different WWAN multicast channels such as Multimedia Broadcast/Multicast Service (MBMS) in 3GPP [6]. A peer interested in a particular view subscribes to the corresponding channel and can switch to an adjacent view interactively by switching multicast channels every T/FPS seconds (an *epoch*), where FPS is playback speed of the video in frames per second.

Peers are also connected to their neighbors via ad hoc WLAN, providing a secondary network for potential CPR frame recovery by relaying each peer's own received packets. If a neighbor to a peer is watching the same view v, then she can assist in frame recovery of same view v by relaying her own received packets via CPR. If neighbor is watching a different view v', then she can still help to partially recover lost frames via view interpolation since the views are correlated.

The WWAN server first multicasts one epoch worth of video to peers. During WWAN transmission of the next video epoch, cooperative peers will exchange received packets or decoded frames of the first video epoch. When the server multicasts the third epoch, peers exchange video packets in the second epoch, and video in the first epoch is decoded and displayed. View-switching delay is hence two epochs $2T/FPS$.

B. Source and Network model

1) Source Model: We assume M views are captured by closely spaced cameras so that strong inter-view correlation exists among them. We assume a Group-of-Pictures (GOP) of a given view, transmitted in one epoch duration T/FPS, is composed of a leading I-frame followed by $T-1$ P-frames; each P-frame F_k is differentially coded using the previous frame F_{k-1} as predictor. A frame F_k is correctly decoded if it is correctly received *and* its predictor (if any) is correctly decoded. A correctly decoded frame F_k reduces visual distortion by d_k. Each frame F_k is divided into r_k packets, $p_{k,1}, \ldots, p_{k,r_k}$, for transmission. The total number of source packets in a GOP is then $R = \sum_{k=1}^{T} r_k$.

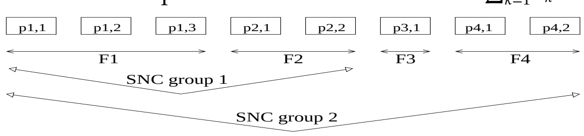


Fig. 2. Example of structured network coding (SNC) for a 4-frame GOP and two SNC groups: $\Theta_1 = \{F_1, F_2\}$, $\Theta_2 = \{F_1, \ldots, F_4\}$.

2) WWAN & Ad hoc WLAN Channel Models: To model WWAN packet losses, we use the Gilbert-Elliot (GE) model with independent & identically distributed (iid) packet loss probabilities g and b for each of 'good' and 'bad' state, and state transition probabilities p and q to move between states.

We assume packets are lost in the ad hoc WLAN due to in-air collision from hidden terminals. Denote by $\gamma_{n,m}$ the probability of a transmitted packet by peer n being lost to a one-hop receiving peer m. For simplicity, we assume they are known and unchanging for the duration of a repair epoch.

Transmissions in the ad hoc WLAN are scheduled according to the 802.11 MAC layer protocol. When the right to send is granted by the MAC layer, a *transmission opportunity* (TO) becomes available to the peer. The peer then decides whether to send and what packet to send during this TO. We assume an acknowledgment (ACK) control packet is broadcasted after receiving a CPR packet and is transmitted without loss.

C. Networking Coding for CPR

In order to improve CPR packet recovery efficiency, it has been proposed [3] that each peer should encode received packets into a coded packet using *network coding* (NC) [7] before performing CPR exchange. More specifically, at a particular TO for peer n, she has received set $\mathcal{G}_n$ of source packets from WWAN streaming source via WWAN and set $\mathcal{Q}_n$ of NC repair packets from neighboring peers via ad hoc WLAN. Peer n can NC-encode a CPR packet, q_n, as a randomized linear combination of packets in $\mathcal{G}_n$ and $\mathcal{Q}_n$:

$$q_n = \sum_{p_{i,j} \in \mathcal{G}_n} a_{i,j} p_{i,j} + \sum_{q_l \in \mathcal{Q}_n} b_l q_l \tag{1}$$

where $a_{i,j}$'s and b_l's are random coefficients for the received source and CPR packets, respectively. This approach is called *Unstructured Network Coding* (UNC). The advantage of UNC is that *any* set of R received *innovative*[1] packets—resulting in R equations and R unknowns—can lead to full recovery of all packets in the GOP. The shortcoming of UNC is that if a peer receives fewer than R innovative packets, then this peer cannot recover *any* source packets using the received CPR packets.

1) Structured Networking Coding for CPR: To address UNC's shortcoming, one can impose structure in the random coefficients $a_{i,j}$'s and b_l's in (1) when encoding a CPR packet, so that partial recovery of important frames in the GOP at a peer when fewer than R innovative packets are received is possible. Specifically, we define X *SNC groups*, $\Theta_1, \ldots, \Theta_X$, where each Θ_x covers a different subset of frames in the GOP and $\Theta_1 \subset \ldots \subset \Theta_X$. Θ_1 is the most important SNC group, followed by Θ_2, etc. Corresponding to each SNC group Θ_x is a *SNC packet type* x. Further, let $g(j)$ be the index of the smallest SNC group that contains frame F_j.

With the definitions above, a SNC packet $q_n(x)$ of type x can now be generated as follows:

$$q_n(x) = \sum_{p_{i,j} \in \mathcal{G}_n} \mathbf{1}(g(i) \leq x) \; a_{i,j} p_{i,j} + \sum_{q_l \in \mathcal{Q}_n} \mathbf{1}(\Phi(q_l) \leq x) \; b_l q_l, \tag{2}$$

where $\Phi(q_l)$ returns the SNC type of packet q_l, and $\mathbf{1}(\mathtt{c})$ evaluates to 1 if clause $\mathtt{c}$ is true, and 0 otherwise. In words, (2) states that a CPR packet $q_n(x)$ of type x is a random linear combination of received source packets of frames in SNC group Θ_x and received CPR packets of type $\leq x$. Using (2) to generate CPR packets, a peer can now recover frames in SNC group Θ_x when $|\Theta_x| < R$ innovative packets of types $\leq x$ are received.

More specifically, we can define the necessary condition to NC-decode a SNC group Θ_x at a peer as follows. Let c_x be the sum of received source packets $p_{i,j}$'s such that $g(i) = x$, *and* received CPR packets of SNC type x. Let C_x be the number of source packets in SNC group x, i.e. $C_x = \sum_{F_k \in \Theta_x} r_k$. We can then define the number of *type x innovative packets* for SNC group Θ_x, I_x, recursively as follows:

$$\begin{aligned} I_1 &= \min(C_1, c_1) \\ I_x &= \min(C_x, c_x + I_{x-1}) \end{aligned} \tag{3}$$

(3) states that the number of type x innovative packets I_x is the smaller of i) C_x, and ii) c_x *plus* the number of type $x-1$ innovative packets I_{x-1}. A SNC group Θ_x is decodable only if $I_x = C_x$.

III. Formulation

We now address the packet selection problem when a peer is granted a TO by the MAC layer: should she send a CPR packet, and if so, which SNC type should the packet be NC-encoded in? We discuss how we solve this problem via a carefully constructed MDP with finite horizon in this section.

[1]A new packet is innovative for a peer if it cannot be written as a linear combination of previously received packets by the peer.

A. Preliminaries

We assume that at the start of the CPR repair epoch of T/FPS seconds, each peer n already knows who are her 1-hop neighbors, and who lost what source packets during WWAN transmission in the last epoch. Among her 1-hop neighbors, those who did not have all their source packets received in the last WWAN broadcast epoch are marked *target receivers*. At each MDP invocation (when a peer is granted a TO by the MAC layer), one peer m out of the pool of marked target receivers is selected (in a round robin fashion for fairness). The objective of the MDP is to maximize the expected distortion reduction of the selected target receiver m. At each TO, a peer can estimate the number of TOs she has left until the end of the repair epoch based on the observed time intervals between previous consecutive TOs, and the amount of time remaining in the repair epoch. Let the estimated number of remaining TOs until the end of the repair epoch be H; hence H is also the finite horizon for the constructed MDP.

Finally, we assume that when a peer received a CPR packet from her neighbor, she immediately transmits a *rich* ACK packet, revealing her current state (the number of CPR packets she has received in each SNC type).

B. State & Action Space for MDP

In a nutshell, a MDP of finite horizon H is a H-level deep recursion, each level t is marked by its states s_t's and actions a_t's. Each state s_t represents the state of the target receiver m $t-1$ TOs into the future, and a_t's are the set of feasible actions that can be taken by the sender at that TO given receiver's state s_t. The solution of a MDP is a *policy* π that maps each s_t to an action a_t, i.e., $\pi : s_t \rightarrow a_t$. We first define states

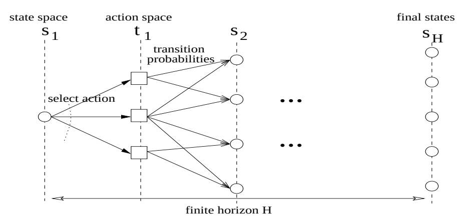


Fig. 3. Example of Markov Decision Process

s_t's for our MDP construction for SNC packet selection. Let $s_t(I_1, ..., I_X, I_1^{'}, ..., I_X^{'}, I_1^{''}, ...I_X^{''})$ be a feasible state for target receiver m at TO t, where I_x is the number of type x innovative packets of the same view as the target receiver. $I_x^{'}$and $I_x^{''}$ are the number of type x innovative packets of the left and right adjacent views to the target receiver. Given $I_x \leq C_x$, the size of the state space is bounded by $O((\prod_{x=1}^{X} C_x)^3)$. In practice, the number of SNC groups X is small, hence the state space size is manageable.

Given each peer receives one view from the WWAN source, the action space for each sender is: i) no transmission ($a_t = 0$), and ii) transmission of CPR packet of type x ($a_t = x$) of the sender's selected view. A type x CPR packet will not be transmitted if there are already sufficient packets to decode SNC group x of that view at receiver. Thus, an action $a_t = x$ is feasible iff the following two conditions are satisfied:

1) There exists source packets $p_{i,j} \in \mathcal{G}_n$ such that $g(i) = x$ and/or $q_m \in \mathcal{Q}_n$ such that $\Phi(q_m) = x$.
2) $I_x < C_x$.

The first condition ensures there is new information pertaining to SNC group Θ_x that can be encoded, while the second condition ensures the encoded packet of type x is not unnecessary.

C. State Transition Probabilities for MDP

State transition probabilities is the likelihood of reaching state s_{t+1} in next TO $t+1$ given state s_t and action a_t at current TO t. Here, we arrive at the "distributed" component of the packet selection problem: the probability of arriving in state s_{t+1} depends not only on the action a_t taken by this peer n at this TO t, but also actions taken by other peers transmitting to the target receiver m during the time between TO t and TO t+1. However, given packet selection is done by individual peers in a distributed manner (as opposed to centralized manner), how can this peer know what actions will be taken by other transmitting peers in the future?

Here, we leverage on previous work in distributed MDP [8] that utilizes the notion of users' aggregate *behavior patterns* in normal-form games. The idea is to identify the patterns of users' tendencies to make decisions (rather than specific decisions), and then make prediction of users' future decisions based on these patterns. For our specific application, first, we assume the number of received packets of the same, left and right adjacent views at target receiver m from other transmitting peer(s) between two TOs are L, $L^{'}$ and $L^{''}$, respectively. These can be learned from target receiver m's ACK messages overheard between the sender's consecutive TOs.

For given L, $L^{'}$ or $L^{''}$ received packets of the same, left or right adjacent view from other sender(s), we identify the corresponding SNC packet types by considering the following two aggregate behavior patterns. First is *pessimistic* and assumes the aggregate of other transmitting peers of this view always transmit innovative packets of the smallest SNC groups possible. This pattern is pessimistic because it seeks immediate benefit as quickly as possible, regardless of the number of TOs available in the finite horizon of H levels. The second is *optimistic* and assumes the aggregate of other transmitting peer(s) will always transmit innovative packets of the largest SNC group Θ_X. This is optimistic because it assumes R innovative packets for the largest SNC group Θ_X will be received by the target receiver m, so that the entire GOP can be correctly decoded.

Let λ, $\lambda^{'}$ and $\lambda^{''}$ be the probabilities that a peer uses pessimistic pattern when selecting a SNC packet type of the same, left and right adjacent views, respectively. The probability that L packets of the same view are divided into k packets of pessimistic and $L-k$ packets of optimistic patterns is:

$$P(k|L) = \binom{L}{k} \lambda^k (1-\lambda)^{L-k} \tag{4}$$

(4) can also be used for the probability $P(k|L^{'})$ or $P(k|L^{''})$ that $L^{'}$ or $L^{''}$ packets are divided into k pessimistic and $L^{'}-k$ or $L^{''}-k$ optimistic packets, with $\lambda^{'}$ or $\lambda^{''}$ replacing λ in (4).

Initially, we do not know which pattern is more likely, and we assume they are equally likely with probability 1/2. The probabilities of pessimistic pattern for the three views, λ, $\lambda^{'}$ and $\lambda^{''}$, will be learned from ACK messages from target receiver m as the CPR process progresses, however.

To derive state transition probabilities, we first define G to be a mapping function that, given state s_t, maps k pessimistic and $L-k$ optimistic packets of view v ($v \in \{s, l, r\}$ to denote same, left and right adjacent view of the receiver) into corresponding SNC packet difference vector $\Delta = \{\delta_1, \ldots, \delta_X\}$, i.e. $G(s_t, v) : (k, L-k) \longrightarrow \Delta$. In general, there can be multiple pessimistic / optimistic combinations $(k, L-k)$'s that map to the same Δ. Let $\Delta = s_{t+1}(I_1, \ldots, I_X) - s_t(I_1, \ldots, I_X)$ be the SNC type-by-type packet count difference between state s_{t+1} and s_t for the same view. Similarly, let Δ' and Δ'' be the type-by-type packet count difference between s_{t+1} and s_t for the left and right adjacent views. Further, let $\Delta^+ = s_{t+1} - s_t - \{a_t\}$, which is like Δ, but also accounting for the CPR packet of the same view transmitted by this sender's current action $a_t = x$.

Assuming action of the same view $a_t = x$, $x \geq 1$, the state transition probability $P(s_{t+1}|s_t, a_t = x)$ can now be written:

$$\begin{aligned}&\gamma_{n,m}\left(\sum_{k|(k,L-k)\xrightarrow{G(s_t,s)}\Delta} P(k|L)\right)\left(\sum_{k|(k,L'-k)\xrightarrow{G(s_t,l)}\Delta'} P(k|L')\right)\left(\sum_{k|(k,L''-k)\xrightarrow{G(s_t,r)}\Delta''} P(k|L'')\right)\\ &+(1-\gamma_{n,m})\left(\sum_{k|(k,L-k)\xrightarrow{G(s_t,s)}\Delta^+} P(k|L)\right)\left(\sum_{k|(k,L'-k)\xrightarrow{G(s_t,l)}\Delta'} P(k|L')\right)\left(\sum_{k|(k,L''-k)\xrightarrow{G(s_t,r)}\Delta''} P(k|L'')\right)\end{aligned} \tag{5}$$

(5) states that to arrive at state s_{t+1}, the L, L', L'' packets received from other senders of the same, left and right adjacent views must lead to packet difference vectors Δ, Δ' and Δ'' if transmitted packet of the same view by this peer n is lost (with probability $\gamma_{n,m}$), or lead to packet difference vectors Δ^+, Δ' and Δ'' if transmitted packet by this peer n is delivered successfully (with probability $1 - \gamma_{n,m}$). Similar expression can be derived for the state transition probability if the sender is transmitting CPR packets of left or right adjacent view.

D. Finding Optimal Policy for MDP

The optimal policy π^* is one that leads to the minimum expected distortion (maximum distortion reduction) at the end of the H-level horizon. More specifically, denote by $\pi_t^*(s_t)$ the maximum expected distortion reduction at the end of H-level horizon, given state at TO t is s_t. $\pi_t^*(s_t)$ can be defined recursively: a chosen action a_t at TO t leads to state s_{t+1} with probability $P(s_{t+1}|s_t, a_t)$, as defined in Section III-C, and assuming optimal policy $\pi_{t+1}^*(s_{t+1})$ is performed at TO $t+1$, we have expected benefit $P(s_{t+1}|s_t, a_t)\pi_{t+1}^*(s_{t+1})$. $\pi_t^*(s_t)$ exhaustively searches for the optimal action a_t^* given state s_t:

$$\pi_t^*(s_t) = \begin{cases} \max_{a_t} \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t)\pi_{t+1}^*(s_{t+1}) & \text{if } t < H \\ d(s_t) & \text{o.w.} \end{cases} \tag{6}$$

If state s_t is at the end of the H-level horizon, then no more actions can be taken, and $\pi_t^*(s_t)$ in (6) simply returns the distortion reduction $d(s_t)$ given state s_t. $d(s_t)$ is defined as follows:

$$\sum_{k=1}^{T} d_k\, \mathbf{1}\left(\bigcup_{x=g(k)}^{X} (I_x = C_x)\right) + d'_k\, \mathbf{1}\left(\bigcap_{x=g(k)}^{X} (I_x < C_x)\right) \mathbf{1}\left(\bigcup_{x=g(k)}^{X} (I'_x = C_x) \cap (I''_x = C_x)\right) \tag{7}$$

(7) states that frame F_k can be recovered from CPR packets of the same view (with distortion reduction d_k), if one of SNC groups $\Theta_{g(k)}, \ldots, \Theta_X$ can be correctly NC-decoded. If all SNC groups $\Theta_{g(k)}, \ldots, \Theta_X$ of the same view fail, then F_k can still be partially recovered (with distortion reduction $d'_k < d_k$) from CPR packets of adjacent views, if one of SNC groups $\Theta_{g(k)}, \ldots, \Theta_X$ of *both* left and right adjacent views can be NC-decoded. (6) can be solved efficiently using dynamic programming as done in [8]. Note that the complexity is determined by the finite horizon H and the size of the state space.

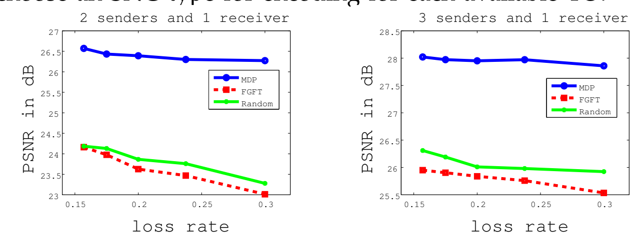
IV. Experimentation

A. Experimental Setup

For our experiments, we used three views of the standard multiview video test sequence `akko` at 640×480 resolution. Each client randomly chose one view for each transmission epoch. A GOP was a leading I-frame plus 9 P-frames. We fixed the quantization parameters for I- and P-frames so that the resulting visual quality in Peak-Signal-to-noise Ratio (PSNR) was roughly 32.5dB. Maximum transport unit (MTU) were assumed to be 1250 bytes, bandwidths of the WWAN broadcast and WLAN were assumed to be 400 and 260kbps, respectively. We assumed there were only two globally defined SNC groups: $\Theta_1 = \{F_1, F_2\}$ and $\Theta_2 = \{F_1, \ldots, F_{10}\}$.

B. Experimental Results

We focused on the case when there was one target receiver and more than one neighboring peer helping to recover lost packets. For comparison, `FGFT` instructs each peer to send CPR packets of SNC group Θ_x only after sending sufficient number of type $x-1$ packets. `random` instructs each peer to randomly choose an SNC type for encoding for each available TO.



(a) $\alpha = 0.1$, $a = 0.04$, $\beta = 0.85$ (b) $\alpha = 0.01$, $a = 0.05$, $\beta = 0.8$

Fig. 4. PSNR comparison of proposed MDP-based decision making and two other schemes for a three/four-node network topology.

In Fig. 4(a), there were two senders and one receiver, and WWAN loss rate was changed by varying loss rate b in B state (x-axis). y-axis is the resulting video quality in PSNR in dB. We can see that `MDP` outperformed `random` and `FGFT` by 2.5dB and 2.5dB, respectively. The main reason for the improvement is that `MDP` maximized expected distortion reduction for the entire repair epoch, taking other transmitting peers' actions into consideration through observed behavior patterns. That means peers can avoid sending duplicate packets. In the case when WLAN loss rates were set to 0.05 and 0.06, one sender could also learn that the other sender had lower WLAN loss rate than herself, and subsequently elected not to transmit during available TOs for better overall system performance. In Fig. 4(b), we change the WWAN GE model parameters to be $\alpha = 0.01$, $a = 0.05$, $\beta = 0.8$ for a four-peer topology. WLAN loss rate was set to be 0.05. We see here that `MDP` outperformed `random` and `FGFT` by 1.8dB and 2.1dB, respectively.

References

[1] J. Crowcroft and K. Paliwoda, "A multicast transport protocol," in *ACM SIGCOMM*, New York, NY, August 1988.

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

[3] X. Liu, G. Cheung, and C.-N. Chuah, "Structured network coding and cooperative wireless ad-hoc peer-to-peer repair for WWAN video broadcast," in *IEEE TMM*, vol. 11, no.4, 2009.

[4] G. Cheung, A. Ortega, and N.-M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," in *IEEE TIP*, vol. 20, no.3, March 2006, pp. 744–761.

[5] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. Springer, 2007.

[6] *Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS) user services; stage 1 (Release 6) (3GPP TS.26.246 version 6.3.0*, March 2006.

[7] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," in *IEEE TIT*, vol. 52, Oct. 2006, pp. 4413–4430.

[8] H. Park and M. van der Schaar, "A framework for foresighted resource reciprocation in P2P networks," in *IEEE Transactions on Multimedia*, vol. 11, no.1, January 2009, pp. 101–116.