

A HYBRID SENDER/RECEIVER-DRIVEN ERROR PROTECTION SCHEME FOR RELIABLE P2P SCALABLE VIDEO STREAMING

Chi-Wen Lo¹, Chia-Wen Lin², and Yung-Chang Chen²

¹Information and Communications Research Labs, Industrial Technology Research Institute, Taiwan

²Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan

E-mail: kenlo0425@gmail.com, {cwwlin, ycchen}@ee.nthu.edu.tw

ABSTRACT

This paper proposes a hybrid sender/receiver-driven error protection scheme to transmit scalable video packets over packet-lossy peer-to-peer networks. In our scheme, given an estimated system uplink capacity, a joint source-channel coding (JSCC) mechanism based on receiver-driven subscriptions is proposed to minimize the visual distortion received by child-peers by subscribing to appropriate amounts of source and channel coding packets. Because the bandwidth for inter-peer transmissions may fluctuate largely due to peer dynamics, in our method peers estimate the available system uplink capacity based on consensus propagation to avoid the fluctuating allocations of JSCC. To efficiently utilize the uplink bandwidth of peers, parent-peers use sender-driven contribution-guided peer selection to reject the low-contribution subscriptions requested from candidate child-peers. Simulation results demonstrate that our method significantly improves visual quality, compared to other state-of-the-art schemes. **Index Terms**—peer-to-peer video streaming, scalable video coding (SVC), forward error correction, joint source-channel coding.

1. INTRODUCTION

Due to the fast growing deployment of advanced network and multimedia technologies, video streaming services are able to provide stable quality. The key of a successful video streaming system lies in the video quality perceived by users. One of the major challenges to video streaming services is the packet loss due to transmission error and network congestion. Since current IP-based networks only support best effort delivery, video packets are not well protected. If a video packet cannot be received before its playback time, the reconstructed video quality may be seriously damaged.

There are two classes of methods to overcome the packet loss problem: retransmission-based and FEC-based schemes. Packet-level FEC has proven to be an efficient means for packet loss recovery in P2P video streaming systems. In a packet-level FEC based protection scheme, the channel encoder, such as Reed-Solomon code, encodes the video bitstreams into k data packets and additional $n-k$ redundant packets, denoted as $FEC(n,k)$. On one hand, a receiver can completely recover the original data should at least any k out of n packets be received. On the other hand, $FEC(n,k)$ scheme can only tolerate loss of $n-k$ packets at most.

The performance of FEC codes with different video frame types was analyzed in [1], showing that packet loss in a P2P video streaming system can be mitigated by using unequal error protection (UEP), where video frames of higher importance are assigned with more redundancy to mitigate packet loss. The packet

loss in P2P streaming systems can also be mitigated by multiple-description coding (MDC) [2]. In [3], the packet loss probability and packet loss accumulation in a multi-source tree-based P2P system are analyzed. In [4], the optimized scalable MDC quantizer can achieve higher distortion gain during data delivery over error-prone networks compared to non-optimized quantizers. In our recent work presented in [5], [6], we proposed a model to estimate the packet loss probability in a mesh-based P2P network and a sender-driven peer-selection scheme by which a parent-peer can adaptively select child-peers to transmit redundant packets according to the link packet loss rates, peer dynamics, and packet loss propagation among peers.

In P2P video streaming, the system uplink capacity may vary largely since heterogeneous peers usually have various uplink bandwidths. Scalable video coding (SVC) is designed to transmit video packets over heterogeneous networks, making it suited to be used in P2P video streaming services. With SVC, peers can adapt video quality according to the system uplink capacity.

In the LayerP2P scheme proposed in [7], child-peers prioritize their chunk requests by layers and parent-peers give higher transmission priority in peer selection to those child-peers who have also sent video chunks to the parent-peers as well. In [8], based on the assumption that a child-peer can obtain the available uplink bandwidth information of their parent-peers through gossiping messages, the child-peer executes a parent-peers selection process to maximize its priority sum composed by the importance of layers.

However, the JSCC for P2P streaming systems was not well addressed. In this paper, we propose a receiver-driven JSCC scheme in which peers can maximize their video quality under the limited uplink capacity by selecting the optimum subscriptions for SVC and FEC packets. Due to the distributed nature and peer dynamics of P2P network, peers cannot access the reliable uplink bandwidth therefore the results of the JSCC scheme also become unstable. To tackle this problem, we apply consensus propagation [9], a distributed averaging scheme, to estimate the global system uplink bandwidth capacity rather than to measure the available uplink bandwidth among neighbor peers. In addition, we propose a contribution-guided peer selection, by which a parent-peer can adaptively select child-peers to transmit subscribed packets according to the rate-distortion contribution of the subscriptions from child-peers, to efficiently allocate the parent-peers' available uplink bandwidth.

2. FEC-BASED PACKET PROTECTION SCHEME

Fig. 1 depicts the proposed scheme of packetization with interleaving used in this paper. In our method, the SVC encoder

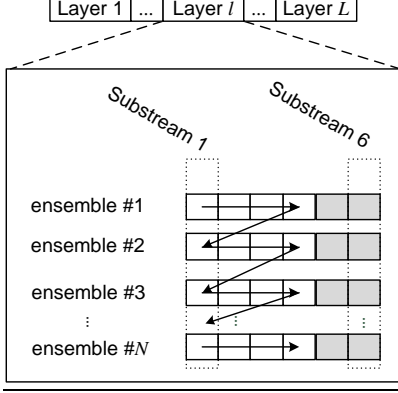


Fig. 1. An FEC packetization example of one SNR scalable layer with FEC(6,4) code, where the white blocks indicate the data packets and the grey ones indicate FEC redundant blocks. The arrows indicate the packet writing direction in our interleaver.

generates L SNR scalable layer bitstreams. Without loss of generality, we assume that k video packets of each layered bitstream compose an FEC coding unit, denoted as an *ensemble*. In order to combat against the burst packet loss due to peer churns, our packet interleaver writes source packets to ensembles in the order indicated by the arrows in Fig. 1. Subsequently, the k video source packets of each ensemble are encoded using the packet-level FEC(n,k) code to generate additional $n-k$ redundant packets. The packets in the same corresponding positions in a set of ensembles compose a video substream. As exemplified in Fig. 1, video substream #1 contains the first packets from ensembles #1 to # N . During a streaming session, the child-peers subscribe to the video substreams (i.e., the pull process) from their parent-peers. Once the parent-peers accept the subscriptions, they continuously push video packets to their child-peers (i.e., the push process), as known as the push-pull methods.

3. JSCC FOR P2P STREAMING

3.1. The Packet Loss Model

Packet loss estimation for P2P networks is more complicated than that for client-server structures. Because the video sources involve multiple peers rather than a single server, packet loss would propagate through the inter-peers transmissions. Moreover, peers will usually unexpectedly join and leave the networks.

For single-layer video streaming, we proposed an analytic model in [6] to estimate the packet loss probability in a mesh-based P2P network. The proposed model takes into account the channel packet drop rate, peer dynamics, and FEC protection to characterize the heterogeneous packet loss behavior of individual video substreams transmitted in an irregular P2P mesh network. For the layered video streaming, since the L -layer scalable video streams are packetized with packet-level FEC independently, the packet loss probability of layer l that peer receives n^l substreams from n^l parent-peers can be expressed by

$$Q^l(n^l) = \sum_{i=0}^{n^l} P_i^{n^l} \cdot q_i, \quad (1)$$

where $P_i^{n^l}$ denotes the probability of i parent-peers leaving the system, which can be modeled by *Continuous-Time Markov Chain*, and q_i denotes the packet loss probability due to departures of i parent-peers. The details of packet loss models can be found in [6].

3.1. Problem Formulation

To minimize the expected distortion at the receiver side, child-peers can subscribe to appropriate amounts of source and channel coding packets under an uplink bandwidth constraint. The source coding rate of layer l is defined as the rate of the k data substreams denoted as $R_{s,l}$ and the channel coding rate of layer l is defined as the rate of the subscribed redundant substreams denoted as $R_{c,l}$.

The proposed JSCC scheme aims to determine the optimum source coding rate $R_{s,l}$ and channel coding rate $R_{c,l}$ of L SNR scalable layers under an estimated system capacity so that the expected distortion of subscribed substreams can be minimized as

$$\begin{aligned} & \text{Min } D_E(R_{s,l}, R_{c,l}) \\ & \text{subject to } \sum_{l=0}^L (R_{s,l} + R_{c,l}) \leq R_{\text{sys}}, \end{aligned} \quad (2)$$

where R_{sys} denotes the estimated system uplink capacity.

Then, the optimal protection level of each layer needs to be determined. We define $\Pi_L = (\pi_0, \pi_1, \dots, \pi_L)$ as the protection level set of L layers and $\pi_l \in (-1, 0, 1, \dots, n-k)$ indicates the protection level for layer l , that is, if the number of subscribed substreams of layer l is less than k , $\pi_l = -1$, and if k data substreams are subscribed, $\pi_l = 0$, and so on. The maximum protection level is thus the total number of redundant substreams, $\pi_l = n-k$. Assume that layer 0 is an initial layer and $\pi_0 = n-k$.

3.3. Recursive Distortion Formulation

To solve the optimization problem in (1), we adopt an efficient representation of expected distortion [10] which recursively characterizes the expected distortion of each layer. The expected distortion of subscribed substreams of L layers can be formulated by

$$\begin{aligned} D_E(R_{s,l}, R_{c,l}) &= D_E(\Pi_L) \\ &= \sum_{l=1}^L \left(\prod_{m=1}^l (1 - Q^m(\pi_m)) \right) \\ &\quad \cdot Q^{l+1}(\pi_{l+1}) \cdot D(\sum_{i=1}^l (u(\pi_i) \cdot R_{s,i})), \end{aligned} \quad (3)$$

where $Q^l(\pi_l)$ denotes the packet loss probability of layer l when the protection level is π_l , as described in (1). $D(\sum_{i=1}^l (u(\pi_i) \cdot R_{s,i}))$ represents the distortion of reconstructed pictures decoded from error-free bitstreams when the source coding rate is $\sum_{i=1}^l (R_{s,i})$ and $u(\cdot)$ is the unit step function.

Denote $\alpha^l = \prod_{m=1}^l (1 - Q^m(\pi_m))$ and $\tilde{r}_{s,l} = \sum_{i=1}^l (u(\pi_i) \cdot R_{s,i})$. Then we can derive the following relations

$$\alpha^{l+1} = \alpha^l \cdot (1 - Q^{l+1}(\pi_{l+1})) \quad (4)$$

$$\alpha^l \cdot Q^{l+1}(\pi_{l+1}) = \alpha^l - \alpha^{l+1} \quad (5)$$

The recursive representation for (3) can be rewritten as

$$D_E(\Pi_L) = \sum_{l=1}^{L-1} (\alpha^l - \alpha^{l+1}) \cdot D(\tilde{r}_{s,l}) + \alpha^L \cdot D(\tilde{r}_{s,L}) \quad (6)$$

$$D_E(\Pi_{L-1}) = \sum_{l=1}^{L-2} (\alpha^l - \alpha^{l+1}) \cdot D(\tilde{r}_{s,l}) + \alpha^{L-1} \cdot D(\tilde{r}_{s,L-1}) \quad (7)$$

Combine (6) and (7), we have

$$\begin{aligned} D_E(\Pi_L) &= D_E(\Pi_{L-1}, \Pi_L) \\ &= D_E(\Pi_{L-1}) + \alpha^{L-1} \cdot (1 - Q^L(\pi_L)) \cdot (D(\tilde{r}_{s,L}) - D(\tilde{r}_{s,L-1})) \end{aligned} \quad (8)$$

In general, the expected distortion when sending l layers with $1 \leq l \leq L$ can be recursively calculated by

$$\begin{aligned} D_E(\Pi_l) &= D_E(\Pi_{l-1}, \Pi_l) \\ &= D_E(\Pi_{l-1}) + \alpha^{l-1} \cdot (1 - Q^l(\pi_l)) \cdot (D(\tilde{r}_{s,l}) - D(\tilde{r}_{s,l-1})) \end{aligned} \quad (9)$$

TABLE I
TRELLIS-BASED SEARCH ALGORITHM

1. Let $N_t = k$. Create the element $\phi_{N_t} = (\Pi_{(L)}^{N_t}, N_t, D_E(\Pi_{(L)}^{N_t}))$ where $\Pi_{(L)}^{N_t} = \underbrace{(0, -1, \dots, -1)}_L$.
2. For each layer, search for elements ϕ_{N_t} having the same value of N_t .
3. For each searched ϕ_{N_t} , if $\pi_l \geq 0$, then increase π_l by 1, i.e., $\Pi_{(L,l)}^{N_t+1} = (\dots, \pi_l + 1, \dots)$ and its expected distortion is $D_E(\Pi_{(L,l)}^{N_t+1})$. Create an element $\phi_{N_t+1}^l = (\Pi_{(L,l)}^{N_t+1}, N_t + 1, D_E(\Pi_{(L,l)}^{N_t+1}))$ in Φ .
4. If $\pi_l = -1$, then increase π_l by 1, and increase N_t by k . Then $\Pi_{(L,l)}^{N_t+k} = (\dots, \pi_l + 1, \dots)$ and the expected distortion is $D_E(\Pi_{(L,l)}^{N_t+k})$. Create an element $\phi_{N_t+k}^l = (\Pi_{(L,l)}^{N_t+k}, N_t + k, D_E(\Pi_{(L,l)}^{N_t+k}))$ in Φ .
5. Select the minimal expected distortion from the elements $\phi_{N_t+1}^l$ in Φ . The selected element $\phi_{N_t+1}^l$ is copied into Φ^* .
6. If $N_t = N_{\text{sys}}$ then terminate the algorithm. Otherwise, $N_t = N_t + 1$, then go to step 2.

3.4. Optimal JSCC

We propose a trellis-based searching algorithm which iteratively searches for the optimum protection level set as formulated in (2). Let N_{sys} be the number of substreams which can be subscribed under the constraint of estimated system capacity, i.e., $N_{\text{sys}} = R_{\text{sys}}/R_{\text{sub}}$, where R_{sub} represents the rate of each substream. Let Φ be the set of trellis elements, $\phi_N = (\Pi_{(L)}^N, N, D_E(\Pi_{(L)}^N))$ where $\Pi_{(L)}^N$ denotes the protection level set consisting of the N subscribed substreams, and $D_E(\Pi_{(L)}^N)$ denotes the expected distortion of protection level set $\Pi_{(L)}^N$. Set N_t as the target number of subscribed substreams in each iteration. Let Φ^* be the optimal protection level sets containing elements $\phi_{N_t}^* |_{N_t=1, \dots, N_{\text{sys}}}$ which have the minimal distortion among elements ϕ_{N_t} with the identical value of N_t . The optimal protection level sets can be determined by the trellis-based search algorithm shown in TABLE I. In step 3 of Table I, $\phi_{N_t+1}^l$ and $\Pi_{(L,l)}^{N_t+1}$ respectively indicate the trellis element and protection level set under which π_l is increased to $\pi_l + 1$ when the target number is changed from N_t to $N_t + 1$. $D_E(\Pi_{(L,l)}^{N_t+1})$ can be recursively calculated from $D_E(\Pi_{(L)}^{N_t})$.

Because child-peers subscribe to the uplink bandwidth in a distributed manner, the available bandwidths of parent-peers may not be fully utilized. Hence, after that peers receive all of the subscribed substreams determined by the trellis-based search algorithm, they can subscribe to one additional substream to further reduce the expected distortion. Similarly, the new protection level set can be determined like step 3: for each layer l , $D_E(\Pi_{(L,l)}^{N_{\text{sys}}+1})$ is calculated, and then the new protection level set with minimal distortion is selected from $D_E(\Pi_{(L,l)}^{N_{\text{sys}}+1})$, $l = 1, \dots, L$. If the additional subscription is accepted then peers can subscribe to the next additional substream.

Due to the distributed attribute of a P2P system, peers may not be able to accurately measure R_{sys} . To address this problem, our method applies the consensus propagation algorithm [9], which is in nature a distributed averaging scheme, to estimate the global system uplink capacity R_{sys} . Each peer only needs to exchange messages to its neighbors without the need of global network topology information.

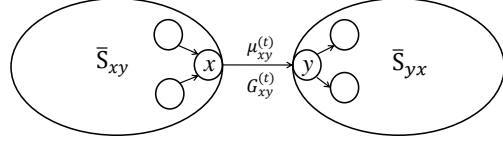


Fig. 2. Illustration of message passing based on consensus propagation.

As depicted in Fig. 2, for any pair of peers $\{x, y\}$, the peers in the set \bar{S}_{yx} can obtain the uplink bandwidth information of the peers in the sets \bar{S}_{xy} through the links between peer x and peer y . To estimate the uplink bandwidth R_{uplink} , the peers in \bar{S}_{yx} must be provided with the average uplink bandwidth message μ_{xy} among the observations at peers in \bar{S}_{xy} and the cardinality $G_{xy} = |\bar{S}_{xy}|$. Similarly, the peers in \bar{S}_{xy} need the information containing the average message μ_{yx} among the observations at peers in \bar{S}_{yx} and the cardinality $G_{yx} = |\bar{S}_{yx}|$. Consider a real-world P2P topology, i.e., loops existing in the network, the messages μ_{xy} and G_{xy} transmitted from peer x to peer y , which can be expressed by

$$\mu_{xy} = \frac{U_x + \sum_{i \in \bar{N}_{x-y}} G_{ix} \mu_{ix}}{1 + \sum_{i \in \bar{N}_{x-y}} G_{ix}} \quad (10)$$

$$G_{xy} = \frac{1 + \sum_{i \in \bar{N}_{x-y}} G_{ix}}{1 + (1 + \sum_{i \in \bar{N}_{x-y}} G_{ix})/\beta} \quad (11)$$

where U_x is the uplink bandwidth capacity of peer x and \bar{N}_x denotes the neighbor peer set of peer x , and $\beta > 0$ is a positive constant. In a loop-free network, the consensus propagation becomes the special case with $\beta = \infty$. Then, each peer can compute the estimated system capacity R_{uplink} by

$$R_{\text{uplink}} = \frac{U_x + \sum_{i \in \bar{N}_x} G_{ix} \mu_{ix}}{1 + \sum_{i \in \bar{N}_x} G_{ix}} \quad (12)$$

3.5. Sender-Driven Contribution-Guided Peer Selection

To improve visual quality, a peer can request source and channel substreams by sending subscription messages to its neighboring peers. Suppose peer y sends a message to peer x to request a substream. Peer x then adds peer y into its candidate set \bar{C}_x . However, due to the limited uplink capacity of peer x , the uplink capacity has to be efficiently allocated to maximize streaming performance. Based on the proposed rate-distortion models, our peer selection method would reject the subscription requests from those ‘‘low-contribution’’ candidate peers whose subscription cannot effectively reduce distortion. Let $N_y + \sigma_{xy}$ be the total number of substreams expected to be received by peer y , where $\sigma_{xy} \in \{1, 0\}$ indicates the peer selection decision for candidate peer y : that is, if peer x selects peer y as a child-peer, $\sigma_{xy} = 1$; otherwise, $\sigma_{xy} = 0$. We use the set of random variables $\bar{\sigma}_x = \{\sigma_{xy} | y \in \bar{C}_x\}$ to represent the set of peer selection decisions.

Consequently, the optimal $\bar{\sigma}_x$ can be determined by minimizing the overall expected distortion of the child-peers, under the uplink bandwidth constraint as follows:

$$\arg \min_{\bar{\sigma}_x} \sum_{y \in \bar{C}_x} D_E(\Pi_L^{N_y + \sigma_{xy}}) \quad (13)$$

subject to $\sum_{y \in \bar{C}_x} \sigma_{xy} \cdot R_{\text{sub}} \leq U_x$ and $\sigma_{xy} \in \{1, 0\}$, where U_x stands for the available uplink bandwidth of peer x . This optimization problem that can be solved by dynamic programming.

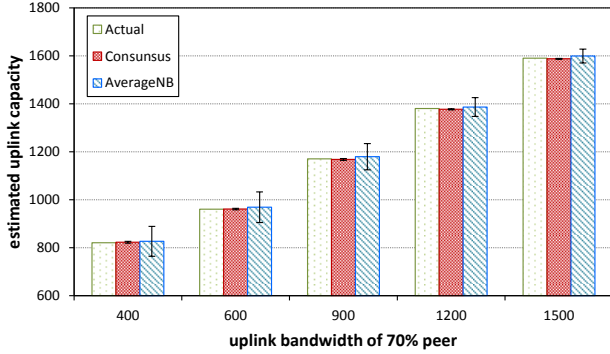


Fig. 3. Comparison of uplink estimation accuracies of two uplink capacity estimation schemes heterogeneous networks.

4. SIMULATION RESULTS

We used P2Pstrsim [11] to simulate a self-organized mesh P2P network of 1000 peers. The whole simulation time period is 30 min. The peer joining rate is 33.3 peers/min and the observation period of peers ranges 15 min to 45 min. We encoded a 300-frame CIF *Foreman* video into 3 SNR scalable layers at 30 fps with a bit-rate of 300 kbps (for each layer) using JSVM 9.15 SVC coder. For each layer, the encoded bitstream is divided into k substreams, each being further divided into fixed-length packets of 1250 bytes. Each ensemble is encoded with FEC(n, k) code, therefore containing n packets, including k packets from the k substreams, and $n-k$ redundant packets composing $n-k$ redundant substreams, where $n = 8$ and $k = 4$ in our experiments. The peers send out subscription messages to request their unavailable substreams once every data scheduling period of 3 s.

To evaluate the performance of three P2P streaming schemes in a heterogeneous network, the peer distribution is composed by 1) 30% peers with 1800 kbps uplink bandwidth, and 2) 70% peers with various uplink bandwidth settings from 400kbps to 1500kbps. The packet loss rate of each link is set to be uniformly distributed in the setting range. We first evaluate the accuracy of uplink capacity estimation schemes with the 95% confidence interval, as depicted in Fig. 3. We compare three results: 1) the actual average uplink capacity, denoted as “Actual,” 2) the uplink capacity estimated by consensus propagation, denoted as “Consensus,” and 3) the uplink capacity estimated by averaging the uplink bandwidths of neighboring peers, i.e., $R_{sys} = \sum_{i \in \bar{N}_x} U_i / |\bar{N}_x|$, denoted as “AverageNB.” As depicted in Fig. 3, the confidence interval with Consensus Propagation are 3.7 kbps, 3.3 kbps, 3.5 kbps, 2.3 kbps, and 1.9 kbps for the uplink bandwidths 400, 600, 900, 1200 and 1500 kbps, respectively. In contrast, the AverageNB scheme leads to significantly larger confidence interval: 62.5 kbps, 63.8 kbps, 57.7 kbps, 39.5 kbps, and 28.8 kbps for the five uplink bandwidths. The reason is that peers average the uplink capacity from their neighbors therefore the accuracy of AverageNB depends on the composition of peers with different uplink bandwidths. However, peers with the Consensus propagation scheme can obtain the accurate estimated uplink capacity averaged from the entire P2P network no matter how they connect to their neighbors.

We then compare the four P2P scalable video streaming schemes: 1) Our JSCC scheme based on the hybrid sender/receiver-driven peer selection (denoted as “Proposed”); 2) Throughput-based scheme [8] (denoted as “Throughput”): Based on the available bandwidth information of their neighbors, child-peers select their parent-peers to maximize the importance of

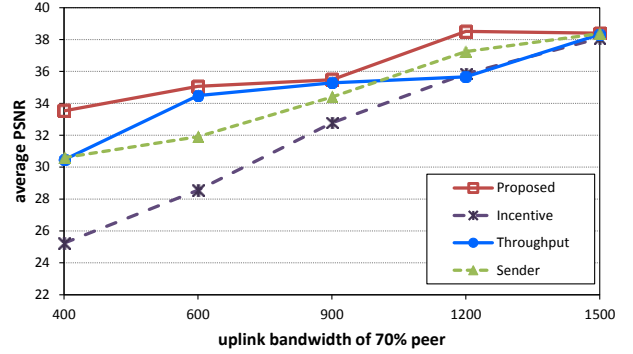


Fig. 4. PSNR performance for peers with various uplink bandwidths under packet drop rate range [1%, 2.5%].

layers; 3) Incentive-based scheme [7] (denoted as “Incentive”): Child-peers categorize their subscriptions into regular and probing subscriptions. The substreams in regular subscriptions are not prioritized among different layers, whereas the substreams in probing subscriptions are requested layer by layer. Parent-peers give higher transmission priority in peer selection to those child-peers who have also sent video chunks to the parent-peers as well; 4) sender-driven peer selection scheme in [6] (denoted as “Sender-Driven”): Child-peers subscribe to substreams layer by layer without R-D optimization. Parent-peers then choose their child-peers according to the child-peers’ contributions described in (13).

Fig. 4 shows the PSNR performance under a link packet drop rate of [1%, 2.5%]. The results show that Incentive and Sender-Driven do not perform well when the uplink bandwidth is lower than 1500 kbps. Note, the proposed method achieves the PSNR quality of 38.5 dB when the uplink capacity is 1200 kbps, whereas the other three schemes require a bandwidth of up to 1500 kbps to achieve the same level of PSNR quality. The reason is that, in the proposed scheme, each child-peer derives an appropriate number of source and channel substreams to subscribe in a rate-distortion optimization manner according to an estimated system uplink capacity. Furthermore, the contribution-guided peer selection can efficiently allocate the uplink bandwidth to the subscribed substreams according to their rate-distortion importance. In contrast, Incentive and Throughput simply adopt layered prioritization (i.e., the lower the layer, the higher the priority) which cannot accurately reflect the rate-distortion importance of substreams. In Sender-Driven scheme, the numbers of received substreams determined by parent-peers in a distributed way are not optimized for child-peers. With the same number of received substreams, our JSCC scheme based on the receiver-driven subscription can minimize the expected visual distortion in child-peers. All schemes can obtain reliable visual quality at 1500 kbps uplink bandwidth which is sufficient to transmit all substreams.

5. CONCLUSION

We proposed a hybrid sender/receiver-driven framework for reliable P2P streaming. First, a JSCC scheme based on receiver-driven subscriptions is used to minimize the distortion of subscribed substreams by properly selecting SVC and FEC redundant substreams based on estimated system capacity. We have also proposed a sender-driven peer selection scheme which rejects low-contribution child-peers to increase bandwidth allocation efficiency under an uplink bandwidth constraint. Simulation results show that our peer selection method achieves significant quality improvement over the compared methods.

6. REFERENCES

- [1] B. Akbari, H. R. Rabiee, and M. Ghanbari, "Packet loss in peer-to-peer video streaming over the Internet," *Multimedia Syst.*, pp. 345–361, 2008.
- [2] V. Padmanabhan, H. Wang, and P. Chou, "Resilient peer-to-peer streaming," *Proc. IEEE ICNP '03*, pp.16–27, Atlanta, GA, Nov. 2003.
- [3] P.-J. Wu, J.-N. Hwang, C.-N. Lee, C.-C. Gau, and H.-H. Kap, "Eliminating packet loss accumulation in peer-to-peer streaming systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 12, pp. 1766–1780, Dec. 2009.
- [4] S. Satti, N. Deligiannis, A. Munteanu, P. Schelkens and J. Cornelis, "Symmetric scalable multiple description scalar quantization", *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3628–3643, July 2012.
- [5] C.-W. Lo, C.-W. Lin, Y.-C. Chen, and J.-Y. Yu, "Contribution-based peer selection for packet protection for P2P video streaming over mesh-based networks," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2011, Brussels, Belgium.
- [6] C.-W. Lo, C.-W. Lin, Y.-C. Chen, and J.-Y. Yu, "Contribution-guided peer selection for reliable P2P video streaming over mesh-based networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 9, pp. 1388–1401, Sept. 2012.
- [7] Z. Liu, Y. Shen, K.W. Ross, S.S. Panwar, and Y. Wang, "LayerP2P: Using layered video chunks in P2P live streaming," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1340–1352, Nov. 2009.
- [8] M. Zhang, Y. Xiong, Q. Zhang, L. Sun, and S. Yang, "Optimizing the throughput of data-driven peer-to-peer streaming," *IEEE Trans. Parallel Distributed Syst.*, vol. 20, no. 1, pp. 97–110, Jan. 2009.
- [9] C.C. Moakkemi, and B.V. Roy, "Consensus propagation," *IEEE Trans. Information Theory*, vol. 52, no. 11, pp.4753–4766, Nov. 2006..
- [10] M. Stoufs, A. Munteanu, J. Cornelis, and P. Schelkens, "Scalable joint source-channel coding for the scalable extension of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 12, pp. 1657–1770, Dec. 2008.
- [11] P2Pstrsim [online].
Available: <http://media.cs.tsinghua.edu.cn/~zhangm/>.