J. Vis. Commun. Image R. 38 (2016) 790–801

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

An unequal error protection scheme for reliable peer-to-peer scalable video streaming ☆

Chi-Wen Lo [a], Chao Zhou [b], Chia-Wen Lin [c,*], Yung-Chang Chen [c]

[a] Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan
[b] Institute of Computer Science and Technology, Peking University, Beijing, China
[c] Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 17 November 2015
Revised 11 March 2016
Accepted 18 April 2016
Available online 29 April 2016

Keywords:
Video streaming
Peer-to-peer streaming
Scalable video coding
Joint source-channel coding
Unequal error protection

ABSTRACT

This paper proposes an unequal error protection (UEP) scheme for transporting scalable video packets over packet-lossy peer-to-peer networks. In our scheme, given an estimated system uplink capacity, a receiver-driven joint source-channel coding (JSCC) mechanism is proposed by which each child-peer minimizes the received visual distortion by subscribing to appropriate numbers of source and channel coding packets. Because the bandwidth for inter-peer transmissions may fluctuate largely due to peer dynamics, in our method, a peer estimates 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 utilize 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 the visual quality, compared to other state-of-the-art schemes.

1. Introduction

Peer-to-peer (P2P) video streaming is an emerging streaming service, which can support large-scale services with a lower infrastructure cost compared to traditional client–server structures. The key of a successful video streaming system lies in the video quality perceived by users. However, one of the major challenges to P2P video streaming services is packet loss. Since current IP-based networks only support best-effort delivery, video packets are not well protected. Moreover, the packet loss problem becomes more serious in P2P video streaming systems because peer dynamics lead to more packet loss opportunities for peers, and the packet loss of a peer will propagate to its neighboring peers through inter-peer transmissions. Such packet loss can seriously damage the quality of reconstructed video.

There are two kinds of methods to overcome the packet loss problem: retransmission-based and FEC-based schemes. In retransmission-based schemes [1,2], a receiver sends a message to a sender to request a lost packet from the sender, and the sender resends the lost packet if the available bandwidth allows. Retransmission-based schemes are particularly useful for non-interactive unicast applications with bursty packet loss. However, since the retransmission-based schemes introduce additional round-trip time latency, they are not suitable for delay-sensitive video transmissions.

Packet-level FEC has proven to be an efficient means for packet loss recovery in P2P video streaming systems [3–7]. 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. XOR-based error correction method was also considered in [8].

In delay stringent streaming environments, FEC based schemes outperform retransmission-based schemes [9]. However, FEC-based video protection schemes would consume additional bandwidth resource to transmit the redundant packets. If the available bandwidth of a P2P system cannot afford the additional amount of redundant packets, packets may get dropped due to traffic congestion. Hence, efficient bandwidth resource utilization for packet-level FEC is desirable.

In P2P video streaming, system channel capacity may vary largely since heterogeneous peers usually have various channel bandwidths. A key technique for achieving bit-rate adaptation is

☆ This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author at: Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan.

E-mail addresses: kenlo0425@gmail.com (C.-W. Lo), zhouchaoyf@gmail.com (C. Zhou), cwlin@ee.nthu.edu.tw (C.-W. Lin), ycchen@ee.nthu.edu.tw (Y.-C. Chen).

http://dx.doi.org/10.1016/j.jvcir.2016.04.014
1047-3203/© 2016 Elsevier Inc. All rights reserved.

scalable video coding (SVC) [10,11] or layered coding. In SVC, the encoder encodes a video into a scalable bitstream which generally contains one base layer (BL) and one or more enhancement layers (ELs). According to the requirements of channel bandwidths of users, a sender can transmit one base layer for basic visual quality or one base layer plus one/more enhancement layers for higher visual quality. SVC is thus a flexible solution to transmitting scalable video contents over heterogeneous networks. With SVC, peers can adapt video quality according to their channel capacity.

Layered coding-based P2P video streaming has been studied recently [12–15]. In the method proposed in [12], child-peers select their parent-peers to maximize its priority sum measured by the importance of layers. In the LayerP2P scheme [13], child-peers categorize their subscriptions into two types: regular subscriptions and probing subscriptions. Child-peers request regular subscriptions, including the substreams of lower layers. The substreams in the regular subscriptions are not prioritized among different layers, whereas those in the probing subscriptions are requested layer by layer. In [14], taxation-based P2P layered streaming designs, including layer subscription strategy, chunk scheduling policy, and mesh topology adaptation, were proposed to adjust the balance between the social welfare and individual peer welfare. Peers can dynamically adjust their subscriptions of layers based on a TCP-style additive-increase-additive-decrease scheme. In [15], layered video data scheduling schemes were proposed to achieve a high delivery ratio of layered video, where the scheduling of requested video blocks was based on the importance factors of video blocks.

To efficiently utilize available bandwidth resource, FEC can protect the layered streams with unequal error protection (UEP). The performance of UEP scheme in FEC-based video streaming can be further improved by using joint source-channel coding (JSCC) [16–21], by which the constrained resource (e.g., uplink bandwidth, transmitted bits) can be optimally allocated between *source coding* (i.e., layered video encoded by SVC) and *channel coding* (i.e., the FEC redundant packets) to minimize the visual distortion at the receiver side. However, JSCC for P2P streaming has not yet been well addressed. Several technical challenges still remain:

1. Packet loss estimation for a P2P network is much more complex than that for traditional client–server structures, since video packets are sourced from multiple peers rather than a single server. Moreover, peers will usually unexpectedly join and leave a system, and such peer churns can cause serious packet loss. Hence, an accurate packet loss model for P2P networks is desirable so that appropriate error protection can be taken to mitigate packet loss.
2. When a parent-peer leaves from or joins in a P2P network, the accessible bandwidths for its child-peers would decrease or increase. Since such peer dynamics make the accessible bandwidth of child-peers from their parent-peers unstable, the allocation results of JSCC schemes would also fluctuate. Instead, peers can estimate the uplink capacity of their parent-peers based on the average capacity of their neighboring peers to avoid short-term fluctuations. However, since the uplink bandwidth of peers is highly heterogeneous in a real-world P2P network, peers may not be able to correctly estimate their parent-peers' average uplink capacity from the statistics of their neighboring peers. The inaccurate bandwidth estimation would lead to incorrect bandwidth allocation between source-coding packets and channel-coding packets.
3. Parent-peers allocate their uplink bandwidth to transmit source-coding packets and channel-coding packets subscribed by child-peers. However, due to the limited uplink capacities of parent-peers, the uplink bandwidth has to be optimally allocated to maximize streaming performance, which is usually difficult considering the irregular inter-peer transmissions in a P2P system.

To address the first problem, a few models for packet loss estimation in P2P networks were proposed [4,5]. In [4], an analytic model was proposed to estimate the packet loss probability of a candidate child-peer in a tree-based P2P network. In a tree-based P2P network, each peer is located in a specific depth in the tree. Therefore, the parent peers of each peer have the same packet loss accumulation if the channel drop rates between peers are homogeneous. In contrast, in a mesh-based P2P network, the peers are randomly located in an irregular mesh structure. As a result, the packet loss accumulation from the multiple parent peers is so heterogeneous that the tree-based packet loss model in [4] cannot correctly characterize the packet loss and propagation behavior. Since many popular P2P streaming systems, such as CoolStreaming [22], PPStreaming [23], and PPLive [24] are based on mesh structures, an accurate mesh-based packet loss model is desirable. In [5], we proposed a model which 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 over the irregular transmission paths in a mesh network.

To estimate the available system capacity of a P2P network, peers can exchange bandwidth information with their neighbors through "consensus algorithms" [25], which reach an agreement of a certain quantity depending on the state of all peers. Consensus propagation [26] was proposed for distributed averaging through simple iterative computation at each node and message exchange among peers. It has found applications in constructing a minimal energy topology in wireless sensor networks [27], where the energy cost of transmission nodes is averaged among nodes. Besides, with consensus propagation, the user rating (e.g., on movies, restaurants) can be shared to P2P social networks [28], where no centralized server collects these rating messages. We shall show in Section 3 that consensus propagation can be used to accurately estimate the average uplink capacity of a P2P network.

To address the issue of parent-peers' uplink bandwidth allocations, a few sender-driven peer selection methods have been proposed in [5,13,29–31]. In sender-driven peer selection, each parent-peer proactively allocates its uplink resource to deliver video data to a selected set of child-peers, that have requested data from the parent-peer, to maximize the overall system performance in throughput, delay, etc. In LayerP2P [13], parent-peers give higher priority in peer selection to those child-peers who have also sent video chunks to the parent-peers in recent historic records. In CoDiO [29], parent-peers schedule the delivery of packets to their child-peers according to the packets' impacts on visual distortion. In addition, those child-peers who have contributed more uplink bandwidth to distribute packets to more succeeding descendants would have higher priorities in packet scheduling and resource allocation for requesting packets from their parent-peers. In the method proposed in [30], parent-peers maximize their available throughput by executing a child-peer selection process. The uplink bandwidth resources are optimally allocated to child-peers based on the available bandwidth of uplink/downlink links and the playback deadline of parent/child-peers. In the rank-based peer selection scheme proposed in [31], a centralized server collects the uplink bandwidth contribution of each peer and gives a rank to each peer according to the peer's bandwidth contribution. Based on the ranking, a parent would serve its child-peers whose ranks are higher than the rank of the parent-peer itself. Therefore the video qualities for high-ranking peers can be guaranteed. In our previous work in [5], a packet loss model is proposed to characterize the packet gains of recovering a lost pack to a peer's descendants. Based on the model, a peer selection scheme is proposed to minimize the overall packet loss rate of a P2P streaming system.

To the best of our knowledge, the JSCC scheme for P2P networks has not been well addressed due to the lack of accurate packet loss models for P2P streaming systems, especially for irregular mesh network structures. In this paper, by extending our previous work [5], we propose a hybrid sender/receiver-driven error protection scheme for reliable SVC-based video streaming over mesh-based P2P networks. The main contribution of this paper is twofold. First, we propose a systematic error protection scheme, involving packet-loss modeling, receiver-driven JSCC, and sender-driven peer selection. In our scheme, given an estimated system uplink capacity, a receiver-driven JSCC mechanism is proposed by which each child-peer minimize its received visual distortion by subscribing to an appropriate amount of source and channel coding packets. To maximize the uplink bandwidth of peers, parent-peers can adaptively select child-peers to transmit subscribed packets according to the rate-distortion contribution of the subscriptions requested from child-peers. Second, we propose a scheme for accurately and stably estimating the parent peers' average uplink capacity based on consensus propagation [26] rather than estimating the average available uplink bandwidth of neighboring peers, which is usually unstable due to the dynamics of neighboring peers and thereby leads to fluctuating allocations of JSCC.

The rest of this paper is organized as follows. The framework of FEC-based error protection is presented in Section 2. The proposed JSCC scheme is described in Section 3. In Section 3.5, the peer selection method is presented. Section 4 shows the simulation settings and the results. Finally, conclusions are drawn in Section 5.

2. Packet protection scheme for P2P streaming

2.1. FEC-based packet protection

In a P2P streaming system, as analyzed in [5], video packets may get lost due to the following three causes: (1) Peer departure: when a parent-peer leaves a system, its child-peers can no longer receive packets from the parent-peer, leading to burst packet loss. The burst packet loss cannot be alleviated until the child-peers find a replacement parent-peer; (2) Link packet loss: the requested packets are dropped due to transmission error or network congestion during transmission; (3) Absent packets in the subscribed parent-peer: a child-peer cannot obtain a packet from its parent-peer if its parent-peer loses the packet. Note, FEC-based packet protection can fully recover lost packets should a sufficient number of FEC packets be received.

Fig. 1 depicts the proposed scheme of packetization with interleaving used in this paper. In our method, the SVC encoder generates an L-layer scalable bitstream. Without loss of generality, we assume that the k fixed-length video packets of each layer compose an FEC coding unit, denoted as an ensemble. In order to combat against burst packet loss due to peer churns, our packet interleaver writes source packets to ensembles in the order indicated by the arrows as shown 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 FEC 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, 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 [22].

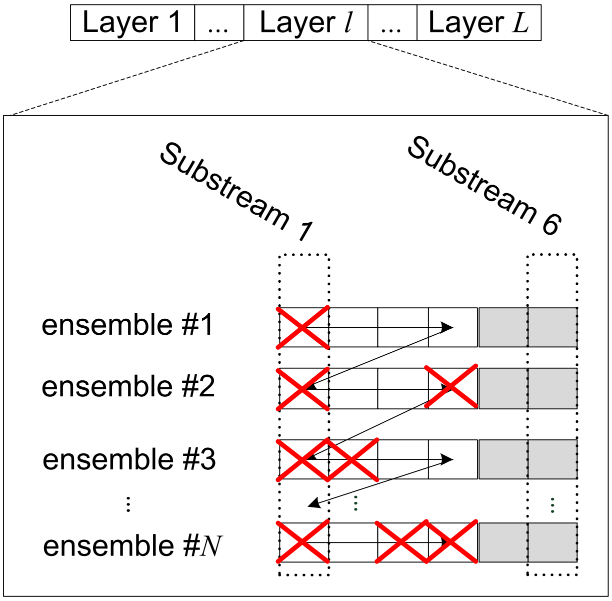

Fig. 1. An FEC packetization example with FEC(6,4) code for L layer, where the white blocks indicate the data packets and the gray ones indicate FEC redundant blocks. The red crosses indicate the lost packets. The arrows indicate the packet writing direction in our interleaver. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In a heterogeneous environment, the packet drop rates of different links can vary largely. Thus different peers require different protection capabilities to successfully recover lost packets. That is, child-peers need FEC codes of different code rates, where the code rate of FEC(n,k) code is k/n. An FEC code with a higher code rate implies lower protection capability. Our method utilizes the punctured Reed-Solomon (RS) code [32] to generate the FEC(n,k) code. With the punctured RS code, peer y requires n_y substreams, $n_y \in (k, k+1, \ldots, n)$ to decode an ensemble by using a single FEC (n,k) decoder instead of several FEC(n_y,k) decoders. Therefore a low-complexity decoder can be used to decode the FEC codes with different code rates k/n_y. As a result, if the number of received packets in an ensemble is at least k, the source packets can be fully recovered with the mother code FEC(n,k). Moreover, packets encoded with punctured RS mother code can be widely exchanged among peers, whereas the packets encoded with different code rates, for example, FEC(6,4) and FEC(8,5), cannot be exchanged. The number of available parent-peers is therefore constrained by the exchangeability of FEC codes of different code-rates.

Fig. 1 illustrates an example of packet loss, where the loss of substream #1 causes a burst packet loss in which the first packets of the ensembles are all lost. When the number of received packets in the same ensemble is more than k, the lost packets can be fully recovered. For ensemble #N, however, the number of received packets is less than k, therefore the lost packets cannot be recovered.

The decoded video quality can be improved by using UEP, by which peers can receive more redundant substreams for important layer to recover the lost packets. However, since the demands for redundant packets may vary largely for heterogeneous peers, a method of determining an appropriate amount of redundancy for each layered stream is desirable.

2.2. The packet loss model

Packet loss estimation for P2P networks is more complicated than that for traditional 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, in a mesh-based P2P network, the peers are randomly located in the irregular mesh structure. Hence, estimating packet loss propagation in a network with an irregular structure (e.g., a mesh network) is much more complex than that in a regular network structure (e.g., a tree network).

For single-layer non-scalable video streaming, we proposed an analytic model in [5] to estimate the packet loss probability in a mesh-based P2P network. Our model takes into account the channel packet drop rate, peer dynamics, and FEC protection to characterize the heterogeneous loss behavior of individual video substreams 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

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

where $P_i^{n^l}$ denotes the probability of i parent-peers leaving the system, which can be modeled by *Continuous-Time Markov Chain* (CTMC), and q_i denotes the packet loss probability model which captures the heterogeneous packet loss behavior of individual video substreams transmitted over the irregular transmission paths of a mesh network. The detailed derivation of (1) can be found in [5].

3. Joint source-channel coding for P2P streaming

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. Define the source coding rate of layer l as the rate of the k data substreams denoted as $R_{s,l}$, and the channel coding rate of layer l as the rate of the subscribed redundant substreams denoted as $R_{c,l}$. The total source coding rate R_{s_sum} and channel coding rate R_{c_sum} are the sums of the rates of layers $0 \sim L$, that is, $R_{s\_\mathrm{sum}} = \sum_{l=0}^{L} R_{s,l}$ and $R_{c\_\mathrm{sum}} = \sum_{l=0}^{L} R_{c,l}$, respectively.

Our JSCC scheme aims to systematically estimate the optimal source coding rates $R_{s,l}$ and channel coding rates $R_{c,l}$ of L layers given the uplink capacity of parent-peers so that the expected distortion of subscribed substreams can be minimized. The JSCC optimization problem can be formulated as

$$\min \ D_E(R_{s\_\mathrm{sum}}, R_{c\_\mathrm{sum}}) \tag{2}$$

$$\text{subject to} \ \sum_{l=0}^{L} (R_{s,l} + R_{c,l}) \leqslant R_{\mathrm{uplink}},$$

where R_{uplink} denotes the estimated uplink capacity.

To solve (2), the optimal protection level of each layer needs to be determined. Define $\Pi_L = (\pi_0, \pi_1, \ldots, \pi_L)$ as the set of protection levels for L layers, where $\pi_l \in (-1, 0, 1, \ldots, n-k)$ indicates the protection level for layer l. That is, if the number of subscribed substreams for layer l is less than k, $\pi_l = -1$; otherwise, 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 the initial layer and $\pi_0 = n - k$.

3.2. Recursive rate-distortion models

In order to solve the constrained optimization problem in (2), we adopt the SVC rate-distortion models proposed in [19] which characterizes the expected distortion of each layer in a recursive manner as follows:

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

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

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

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

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

The recursive representation for (3) can be rewritten as

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

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

Combine (6) and (7), we have

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

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

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

3.3. Optimal JSCC

To search for the optimal set of protection levels formulated in (2), we propose a trellis-based iterative search algorithm. Let N_{sub} be the number of substreams which can be subscribed to under the constraint of estimated system capacity, i.e., $N_{\mathrm{sub}} = R_{\mathrm{uplink}}/R_{\mathrm{sub}}$, where R_{sub} represents the rate of each substream. Let Φ be the set of trellis elements, as depicted in Fig. 2, $\phi_N = \left(\Pi_L^N, N, D_E\left(\Pi_L^N\right)\right)$

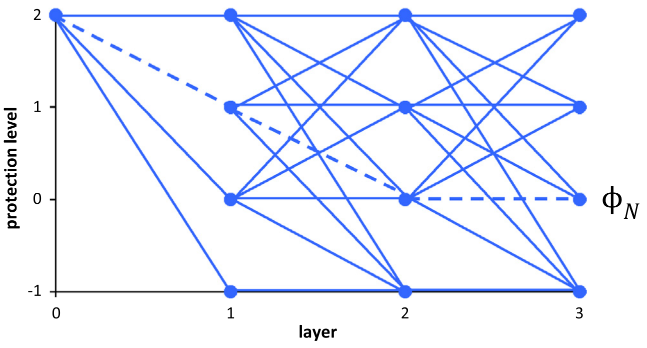

Fig. 2. A three-layer trellis structure involving all possible trellis elements which are indicated by the lines from layer 0 to layer 3. The dash lines indicate an example of trellis elements, namely ϕ_N.

where Π_L^N denotes the set of protection levels for the N subscribed substreams, and $D_E\left(\Pi_L^N\right)$ denotes the expected distortion given Π_L^N. Let N_t denote the target number of subscribed substreams in each interaction, and Φ^* be the optimum protection level sets containing the elements $\phi_{N_t}^*$, $N_t = 1, \ldots, N_{\text{sub}}$ which reach the minimal distortion among the 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 1.

In the beginning, peers need to receive at least k substreams for layer one to obtain a baseline video quality. In step 3 of Table 1, $\phi_{N_t+1}^l$ and $\Pi_{(L,l)}^{N_t+1}$ respectively indicate the trellis element and protection level set under which π_l is increased by one (i.e., $\pi_l + 1$) when the target number is changed from N_t to $N_t + 1$. Moreover, $D_E\left(\Pi_{(L,l)}^{N_t+1}\right)$ can be recursively calculated from the result of $D_E\left(\Pi_L^{N_t}\right)$. As a result, child-peers subscribe to the number of l layer substreams according to $\phi_{N_t}^*$, and then send the subscription messages to the parent-peers who own the requested substreams. The parent-peers then individually select their child-peers based on a sender-driven peer selection scheme, as detailed in Section 3.5.

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

3.4. Consensus propagation

Due to the distributed nature of P2P systems, peers may not be able to accurately measure R_{uplink}. Estimating R_{uplink} by averaging the uplink bandwidths of one peer's neighbors may lead to an estimate biased by the local neighbors, or even a fluctuating estimate due to the dynamics of the local neighbors. The biased or fluctuating estimates of R_{uplink} would lead to improper subscription requests from child-peers. Since parent-peers allocate their available uplink bandwidth through a distributed peer selection scheme, i.e., parent-peers do not corporately select the child-peers, the improper subscriptions will probably be accepted, thereby leading to severe performance degradation. To address this problem, our method utilizes the consensus propagation algorithm [26], which is in nature a distributed averaging scheme, to estimate the global average uplink capacity of parent-peers. With consensus propagation, each peer only needs to exchange messages with its neighbors without the need of collecting the global network topology information.

Table 1
Trellis-based search algorithm.

```
1. Let $N_t = k$. Create the element $\phi_{N_t} = \left(\Pi_L^{N_t}, N_t, D_E\left(\Pi_L^{N_t}\right)\right)$ where
   $\Pi_L^{N_t} = \underbrace{(0, -1, \ldots, -1)}_{L}$
2. For each layer, search for elements $\phi_{N_t}$ having the same value of $N_t$
3. For each searched $\phi_{N_t}$, if $\pi_l \geqslant 0$, then increase $\pi_l$ by 1, i.e.,
   $\Pi_{(L,l)}^{N_t+1} = (\ldots, \pi_l + 1, \ldots)$ and its corresponding expected distortion is
   $D_E\left(\Pi_{(L,l)}^{N_t+1}\right)$. Create an element $\phi_{N_t+1}^l = \left(\Pi_{(L,l)}^{N_t+1}, N_t + 1, D_E(\Pi_{(L,l)}^{N_t+1})\right)$ in $\Phi$
4. On the other hand, if $\pi_l = -1$, then increase $\pi_l$ by 1, and increase $N_t$ by $k$.
   Then $\Pi_{(L,l)}^{N_t+k} = (\ldots, \pi_l + 1, \ldots)$ and the corresponding expected distortion
   is $D_E\left(\Pi_{(L,l)}^{N_t+k}\right)$. Create an element $\phi_{N_t+k}^l = \left(\Pi_{(L,l)}^{N_t+k}, N_t + k, D_E(\Pi_{(L,l)}^{N_t+k})\right)$ in $\Phi$
5. Select the minimal expected distortion from the elements $\phi_{N_t+1}^l$ in $\Phi$. The
   selected element $\phi_{N_t}^{*l}$ is copied into $\Phi^*$
6. If $N_t = N_{\text{sub}}$ then terminate the algorithm. Otherwise, $N_t = N_t + 1$, then go
   to step 2
```

As depicted in Fig. 3, for any pair of peers $\{x,y\}$, the peers in set $\overline{\mathrm{S}}_{yx}$ can obtain the uplink bandwidth information of the peers in sets $\overline{\mathrm{S}}_{xy}$ through the links between peer x and peer y. To estimate the uplink bandwidth R_{uplink}, the peers in $\overline{\mathrm{S}}_{yx}$ must be provided with the average uplink bandwidth message μ_{xy}^* among the observations at peers in $\overline{\mathrm{S}}_{xy}$ and the cardinality $G_{xy}^* = |\overline{\mathrm{S}}_{xy}|$. Similarly, the peers in $\overline{\mathrm{S}}_{xy}$ need the information containing the average message μ_{yx}^* among the observations at peers in $\overline{\mathrm{S}}_{yx}$ and the cardinality $G_{yx}^* = |\overline{\mathrm{S}}_{yx}|$. Consider a simple P2P topology, i.e., no loop existing in the network, the messages $\mu_{xy}^{(t)}$ and $G_{xy}^{(t)}$ transmitted from peer x to peer y at time t can be viewed as iterative estimates of the values μ_{xy}^* and G_{xy}^*. The messages can be expressed by

$$\mu_{xy}^{(t)} = \frac{U_x + \sum_{i \in \overline{\mathrm{N}}_x - y} G_{ix}^{(t)} \mu_{ix}^{(t)}}{1 + \sum_{i \in \overline{\mathrm{N}}_x - y} G_{ix}^{(t)}} \tag{10}$$

$$G_{xy}^{(t)} = 1 + \sum_{i \in \overline{\mathrm{N}}_x - y} G_{ix}^{(t)} \tag{11}$$

where U_x denotes the uplink bandwidth capacity of peer x and $\overline{\mathrm{N}}_x$ denotes the neighbor peer set of peer x.

At time t, each peer can compute the estimated system capacity R_{uplink} by

$$R_{\text{uplink}}^{(t)} = \frac{U_x + \sum_{i \in \overline{\mathrm{N}}_x} G_{ix}^{(t)} \mu_{ix}^{(t)}}{1 + \sum_{i \in \overline{\mathrm{N}}_x} G_{ix}^{(t)}}. \tag{12}$$

Since the distance between peer x and any peer in $\overline{\mathrm{S}}_{xy}$ is smaller or equal to the diameter of P2P network, if t is large enough to allow the message $\mu_{xy}^{(t)}$ and $G_{xy}^{(t)}$ to traverse the diameter of P2P network, then the message will contain the uplink bandwidth capacities of all peers. Thus, we have $\mu^{(t)} = \mu^*$ and $G^{(t)} = G^*$ and (12) becomes

$$R_{\text{uplink}}^{(t)} = \frac{U_x + \sum_{i \in \overline{\mathrm{N}}_x} G_{ix}^* \mu_{ix}^*}{1 + \sum_{i \in \overline{\mathrm{N}}_x} G_{ix}^*} = R_{\text{uplink}}. \tag{13}$$

As a result, $R_{\text{uplink}}^{(t)}$ will converge to the global average R_{uplink}.

In a real-world P2P mesh network, peers may constitute a loop in which a peer traverses back to itself through one or multiple hops along different substream delivery paths. In this case, $G_{xy}^{(t)} \to \infty$ and the algorithm does not converge. We can use a heuristic that adds an attenuation term in (11), that is

$$\tilde{G}_{xy}^{(t)} = 1 + \sum_{i \in \overline{\mathrm{N}}_x - y} G_{ix}^{(t-1)} \tag{14}$$

$$G_{xy}^{(t)} = \frac{\tilde{G}_{xy}^{(t)}}{1 + \tilde{G}_{xy}^{(t)} / \beta}. \tag{15}$$

where $\beta > 0$ is a positive constant. In a loop-free network, the consensus propagation becomes the special case with $\beta = \infty$.

In contrast, in a loop-prone network, $G_{xy}^{(t)}$ becomes increasingly attenuated as $\tilde{G}_{xy}^{(t)}$ grows. Furthermore, the choice of β is critical to the convergence time and accuracy. However, the optimal

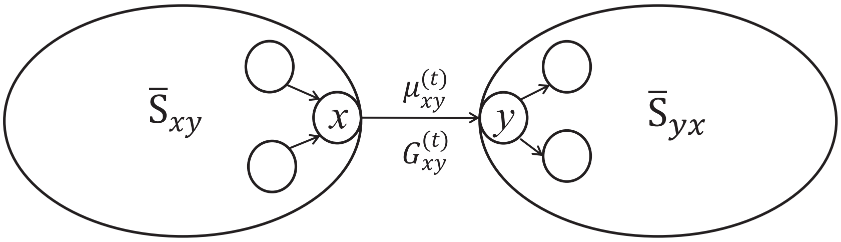

Fig. 3. Illustration of distributed averaging scheme based on consensus propagation.

choice is difficult to determine because the global topology information is required to minimize the convergence time. Hence, an empirical search method can be applied to decide on the appropriate setting of β. According to our experimental results, the choice of $\beta = 2$ can provide stable consensus results. The consensus propagation algorithm is summarized in Table 2. Peers will continuously execute the consensus propagation algorithm until they leave the P2P streaming system.

The messages μ_{xy}^{*} and G_{xy}^{*} can be exchanged between neighbor-peers through gossip messages, which consume two 4-byte overhead to transmit μ_{xy}^{*} and G_{xy}^{*}, respectively. Suppose each peer exchanges gossip message every one second with 100 neighbor-peers. The overhead for the consensus messages is 6.4 kbps which is almost negligible compared to video data rate.

3.5. Sender-driven contribution-guided peer selection

In order to increase the visual quality, a peer can subscribe to source-coding substreams and channel-coding 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 child-peer set $\overline{C}_x$. However, due to the limited uplink capacity of peer x, the uplink capacity has to be efficiently allocated to selected child-peers 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 assist in reducing 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 $\bar{\sigma}_x = \left\{\sigma_{xy} | y \in \overline{C}_x\right\}$ to represent the set of peer selection decisions. Based on the proposed subscription scheme, child-peer y sends a subscription message to parent-peer x to request one substream of layer l, and then the protection level for layer l in child-peer y is changed from π_l to $\pi_l + \sigma_{xy}$. The expected distortion after receiving the substream becomes

$$D_E\left(\Pi_L^{N_y+\sigma_{xy}}\right) = D_E(\ldots, \pi_l + \sigma_{xy}, \ldots). \tag{16}$$

Table 2
Consensus propagation algorithm.

1. At time $t = 0$, $G_{xy}^{(t)} = 0$, $\mu_{xy}^{(t)} = U_x$
2. At time $t - 1$, peer y stores the recent received messages $\left\{\mu_{xy}^{(t-1)}, G_{xy}^{(t-1)} | x \in \overline{N}_y\right\}$ from each neighboring peer
3. At time t, peer y calculates the messages $\left\{\mu_{yz}^{(t)}, G_{yz}^{(t)} | z \in \overline{N}_y\right\}$ and forwards the messages to its neighboring peers. Based on the messages, peer y calculates $R_{\text{uplink}}^{(t)}$
4. In the next time interval, $t = t + 1$, go to step 2

In our peer selection scheme, when a candidate child-peer requests a substream from a parent-peer, the parent-peer will evaluate the candidate peer's contribution in distortion reduction. Suppose parent-peer x selects its child-peers once every peer selection period. During a peer selection period, the child-peers who already requested video substreams from peer x are added into the candidate set of peer x. Besides, those child-peers who have supplied the substreams of a video session and have connected to peer x for more than T seconds are added into the candidate set as well. Note, the connection time to a substream source is set to be at least T seconds to avoid frequent switching (a.k.a. the oscillation effect) among different substream sources. Consequently, the optimal $\bar{\sigma}_x$ is determined by minimizing the overall expected distortion of the child-peers, under the uplink bandwidth constraint as follows:

$$\arg\min_{\bar{\sigma}_x} \sum_{y \in \overline{\mathbf{C}}_x} D_E\left(\Pi_L^{N_y+\sigma_{xy}}\right) \tag{17}$$

$$\text{subject to} \sum_{y \in \overline{C}_x} \sigma_{xy} \cdot R_{\text{sub}} \leqslant U_x \text{ and } \sigma_{xy} \in \{1,0\}$$

Eq. (17) is equivalent to

$$\underset{\bar{\sigma}_x}{\text{argmax}} \sum_{y \in \overline{C}_x} \left\{w_y \cdot \sigma_{xy} \cdot \Delta D_y\right\} \tag{18}$$

$$\text{subject to} \sum_{y \in \overline{C}_x} \sigma_{xy} \cdot R_{\text{sub}} \leqslant U_x \text{ and } \sigma_{xy} \in \{1,0\}$$

where $\Delta D_y = D_E(\ldots, \pi_l, \ldots) - D_E(\ldots, \pi_l + 1, \ldots)$. Note, this optimization problem is kind of 0/1 knapsack problem which can be solved by dynamic programming [33]. Note, the bandwidth utilization can be further improved by selecting $\sigma_{xy}, 0 \leqslant \sigma_{xy} \leqslant 1$. In this case, the child-peers will only receive the substream partially, and then have to request the remaining of the substream, leading to more complicated system design and unstable visual quality.

3.6. Complexity analysis

In the proposed trellis-based search algorithm presented in Table 1, a peer searches for the optimal protection level set composed of N_t substreams by adding one protection level to each layer from the searched protection level sets composed of $N_t - 1$ substreams. Hence, the protection level set will be increased by at most L at one iteration. In the next iteration, the L new protection level sets are calculated from each protection level set generated in the previous iteration. Therefore, there will be at most L^L calculations in one iteration. Since the target number of substream is N_t, there will be N_t iterations in the algorithm. Therefore, the computation complexity is $O(N_t L^L)$. Note, in practical scalable video application, the number of layers L is a constant and usually does not exceed 5. Hence, the complexity of the search algorithm is not high.

Each parent-peer uses (18) to select the high-contribution child-peers to forward substream packets. Since $|\overline{C}_x|$ is the number of candidate child-peers of peer x and $b = U_x/R_{\mathrm{sub}}$ is the number of substream can be served by peer x, the complexity of (18) is $O\left(|\overline{\mathrm{C}}_x| \cdot U_x/R_{\mathrm{sub}}\right)$. In the proposed scheme, each video bitstream is divided into L layers and each layer is further encoded with FEC (n,k) to generate n substreams. Hence, each child-peer will subscribe $L \cdot n$ substreams at most. Assume a P2P network has $|\overline{\mathrm{N}}_{\mathrm{live}}|$ live peers, where $\overline{\mathrm{N}}_{\mathrm{live}}$ is the live peer set. There will be $|\overline{\mathrm{N}}_{\mathrm{live}}| \cdot L \cdot n$ requests that need to be served by $|\overline{\mathrm{N}}_{\mathrm{live}}|$ parent-peers. Hence, each parent-peer on average receives $L \cdot n$ requests from $L \cdot n$ candidate child-peers, i.e., $|\overline{\mathrm{C}}_x| \approx L \cdot n$. In real-world applications, the number of L usually ranges from 3 to 5, and the choices of n and k are few (say, less than 10) to reduce decoding complexity. Besides, the allocation bandwidth unit $R_{\mathrm{sub}} = R_{s,l}/k$ is usually not too small. As a result, the computational complexity of (18) for finding the optimal $\bar{\sigma}_x$ in parent-peers is affordable.

4. Experimental results

We used P2Pstrsim [34] to evaluate the accuracies of uplink bandwidth estimation and the performances of error protection schemes. We used GT-ITM [35] to generate a topology with 1000 peers as the configuration of simulations with P2Pstrsim. The end-to-end delay between two peers is set to be distributed in the range of 10–500 ms uniformly. The simulation time is set to 30 min, where peers uniformly join the network within 30 min, and then leave the system independently after mean user viewing time T_v, which is assumed to be uniformly distributed in the range of $T_v/2$–$3T_v/2$. We set $T_v = 30$ min in our simulations, thus making the viewing time of peers distribute in the range of 15–45 min uniformly. We also assume each user records the number of parent-peer departures in a time period and the average time to find a new replacement parent-peer so as to calculate the peer dynamic parameters used in our packet loss models [5].

We encoded four video sequences including *Foreman*, *Football*, *City*, and *Ice* into 3 SNR scalable layers using the Joint Scalable Video Model (JSVM) version 9.15 SVC coder [37]. The resolutions, bit-rates of each layer, and PSNR values of receiving 1–3 layers (i.e., BL, BL + EL-1, BL + EL-1 + EL-2) for the five sequences are listed in Table 3. For each layer, the encoded bitstream is divided into k substreams transmitted using the UDP protocol, each being further divided into fixed-length packets of 1250 bytes. Each ensemble is encoded with FEC(n,k) code, therefore consisting of n packets, including k packets from the k substreams, respectively, and $n-k$ redundant packets that compose the $n-k$ additional substreams, where $n = 8$ and $k = 4$ in our experiments. As a result, peers can choose either of five code-rate FECs [i.e., FEC(n_y,4), $n_y = 4, 5, \ldots, 8$] of different protection capabilities according to the codes' distortion gains. Peers send out subscription messages to request their unavailable substreams once every data scheduling period that is set to 3 s. When the content server sends out the last packet of a video bitstream, it will start to send the first packet of the video again.

Note, our experiments were mainly designed to simulate wireline networks with different levels of congestion, which lead to different link packet drop rates (1–15%). Nevertheless, to address the burst loss problem due to peer churns, our method uses packet-level FEC with interleaving. To tackle the burst packet loss for wireless channel, our rate distortion model (3) can be combined with an existing burst packet loss model [36] to estimate the distortion due to burst packet loss so as to provide efficient packet protection accordingly.

Table 3
Source coding parameters.

Sequence	Resolution	Layer rate	PSNR of receiving 1–3 layers
Foreman	CIF@30fp	300 kbps	30.15 dB, 35.48 dB, 38.91 dB
Football	CIF@30fps	700 kbps	29.29 dB, 33.92 dB, 37.93 dB
City	4CIF@30fps	1600 kbps	29.23 dB, 33.73 dB, 36.34 dB
Ice	4CIF@30fps	1000 kbps	34.31 dB, 38.71 dB, 40.97 dB
SaraAndKeristan	720P@30fps	800 kbps	31.07 dB, 35.94 dB, 41.63 dB

To evaluate the quality scalability of the proposed error protection scheme for scalable video bitstreams in a heterogeneous network, we synthetically set that the peer distribution is composed of 30% peers with 1800 kbps uplink bandwidth, and 70% peers with various uplink bandwidths ranging from 400 kbps to 1500 kbps to simulate a P2P network where peers may have different uplink capacity. The packet drop rate of each link is set to be uniformly distributed in a specified range, e.g., for the peers of [1%, 5%], the link packet drop rate is uniformly distributed in the range of 1–5%.

We first evaluate the accuracy of uplink capacity estimation schemes with the 95% confidence interval, as depicted in Fig. 4. We compare three results: (1) the actual average uplink capacity (denoted as "Actual") calculated by averaging the uplink bandwidths of peers in two classes, (2) the uplink capacity estimated by the consensus propagation algorithm, denoted as "Consensus," and (3) the uplink capacity estimated by averaging the uplink bandwidths of neighboring peers, i.e., $R_{\mathrm{uplink}} = \sum_{i \in \overline{\mathrm{N}}_x} U_i/|\overline{\mathrm{N}}_x|$, denoted as "AverageNB." As depicted in Fig. 4(a), the confidence interval with the Consensus scheme 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, respectively. The reason is that, with the AverageNB scheme, peers estimate the uplink capacity solely based on the statistics of their local neighboring peers, therefore the estimation result can be biased by the local statistics. Particularly,

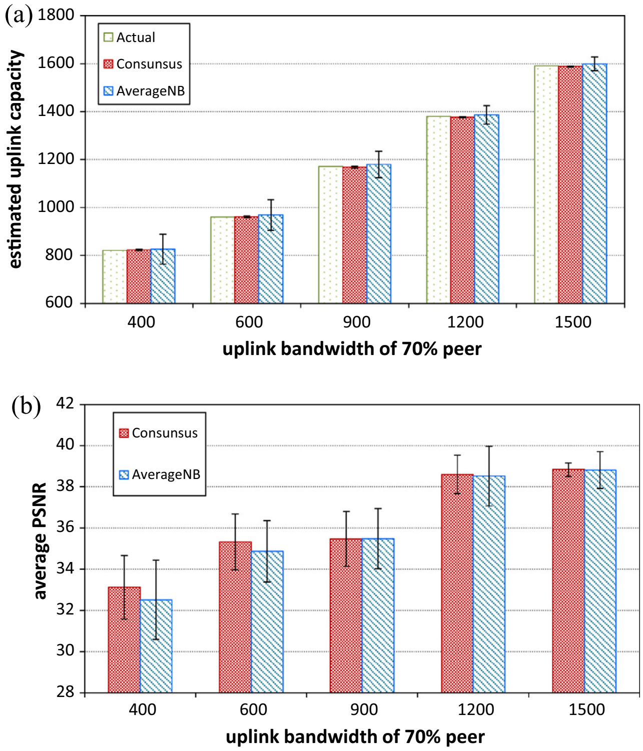

Fig. 4. Performance comparison (Foreman sequence) of two uplink capacity estimation schemes heterogeneous networks: (a) uplink capacity estimation accuracies; and (b) PSNR performances with the two estimation schemes.

under the setting of 70% peers with low uplink bandwidth (i.e., 400 kbps and 600 kbps), the dynamics (i.e., joining and leaving) of the peers with high uplink bandwidth (i.e., 1800 kbps) would lead to higher variation on the estimated uplink capacity of their neighbor-peers. However, thanks to the message propagation used in the Consensus scheme, peers would gain much better knowledge about the global capacity of the P2P network so as to obtain significantly accurate estimates of the network's uplink capacity no matter how they connect to their neighbors. Fig. 4(b) shows the PSNR performance of proposed P2P streaming scheme based on the two uplink capacity estimation schemes. The proposed protection scheme with Consensus estimation achieves higher and more stable PSNR performance compared to that with AverageNB, because, with better estimates of uplink capacity, peers would request subscriptions more appropriately, thereby reducing the distortion due to inadequate bandwidth allocation. On the contrary, when AverageNB overestimates the uplink capacity for a peer, the peer would aggressively request subscriptions, and part of the subscriptions will be rejected by its parent-peers due to their limited uplink capacity which will cause visual quality degradation compared to the optimal subscriptions. On the other hand, when AverageNB underestimates the uplink capacity for a peer, the peer can obtain additional substreams by sending excessive trial subscriptions without considering the uplink capacity constraint. However, the trial subscriptions would only reach a local optimum rather than the global optimum that takes into account the overall constraints of uplink capacities for all subscripted substreams. As a result, our protection scheme with AverageNB leads to worse PSNR performance compared to that with consensus propagation.

Fig. 5 shows the performances of the proposed method using different bandwidth estimation intervals for exchanging consensus propagation messages. The simulation settings are the same as Fig. 4 but the uplink bandwidth capacity is set to 400 kbps for 70% peers. As depicted in Fig. 5(a), the bandwidth estimation results of interval 1 s, 5 s, 10 s, 20 s, and 60 s are 822.5 kbps, 809.8 kbps, 801.5 kbps, 750.6 kbps, and 556.7 kbps with confidence interval 19.5 kbps, 43.3 kbps, 46.8 kbps, 105.7 kbps and 151.5 kbps, respectively. Since the neighbors of a child-peer may dynamically join/depart the network, the child-peer will possibly capture out-of-date and inaccurate uplink bandwidth estimates should the estimation time interval is long, leading to visual quality degradation as illustrated in Fig. 5(b). The average PSNR performances for intervals of 5 s, 10 s, 20 s and 60 s are degraded by 0.62 dB, 0.83 dB, 1.15 dB, and 1.5 dB, respectively, compared to that for the interval of 1 s. Although a shorter interval achieves more accurate estimation results and thus better PSNR performance, it also incurs more overhead to transmit μ^*_{xy} and G^*_{xy}. However, as shown in Fig. 5(c), the overhead is almost negligible compared to the video data rate even if the interval is one second. Hence, we set the bandwidth estimation interval to be one second in the following simulations.

Next, we evaluate the performances of the four P2P scalable video streaming schemes: (1) Our JSCC scheme based on the hybrid sender/receiver-driven peer selection: Child-peers subscribe to layered substreams according to the optimal protection set described in (1). Parent-peers then select their child-peers according to the child-peers' contributions in distortion reduction as described in (18); (2) Sender-driven peer selection [5] (denoted as "Sender-driven"): Child-peers subscribe to substreams layer by layer without considering rate-distortion optimization. Parent-peers then select their child-peers according to the child-peers' contributions in distortion reduction as described in (18); (3) Throughput-based streaming scheme [12] (denoted as "Throughput"): Based on the available bandwidth information of their neighbors obtained through the gossiping messages, child-peers select their parent-peers to maximize their utility evaluated based on the importance of layers; (4) Incentive-based streaming scheme [13] (denoted as "Incentive"): Child-peers categorize their subscriptions into two types: regular subscriptions and probing subscriptions. They first request regular subscriptions, including the substreams of lower layers, under an uplink capacity measured from neighboring peers. Then, they send probing subscriptions if their parent-peers have surplus uplink bandwidth allocated to the child-peers. The substreams in the regular subscriptions are not prioritized among different layers, whereas the substreams in the probing subscriptions are requested layer by layer. Parent-peers give higher transmission priority in peer selection to those child-peers who have also contributed in sending video data to the parent-peers as well.

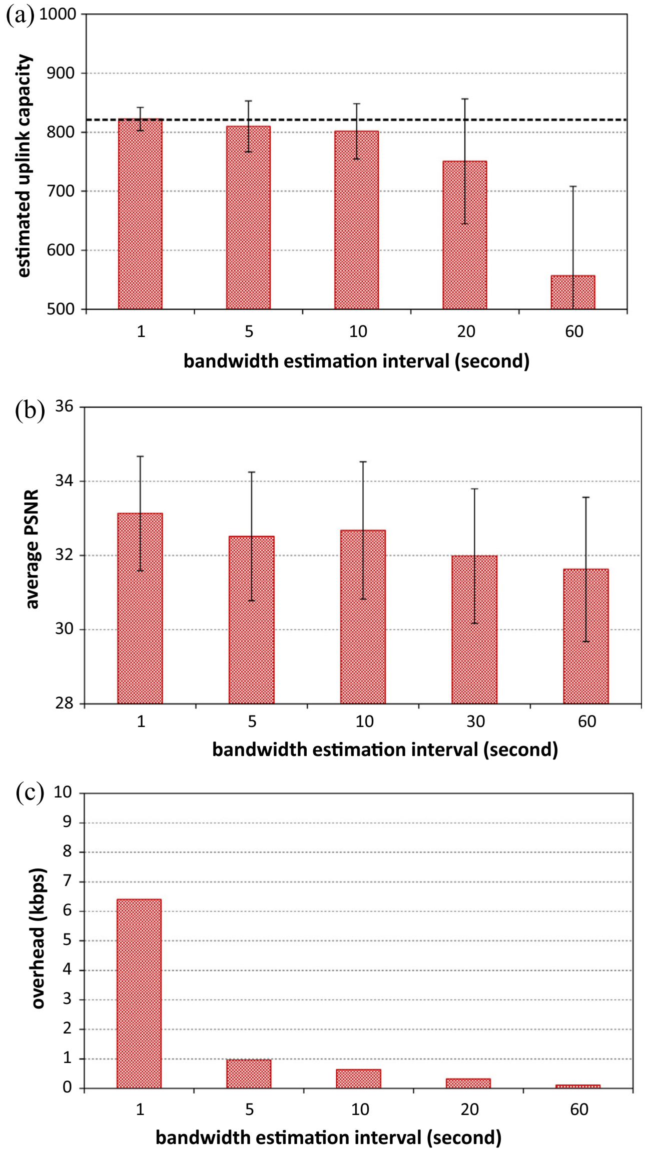

Fig. 5. Performance comparison (Foreman sequence) of the proposed method using different bandwidth estimation intervals for exchanging consensus propagation messages: (a) uplink capacity estimation accuracy; (b) PSNR performances; and (c) overhead. The dot line indicates the ground truth (i.e.: 820 kbps).

Fig. 6 compares the PSNR performances of four P2P streaming schemes under various link conditions. Fig. 6(a) shows the PSNR performance under a low link packet drop rate of [1%, 2.5%]. The results show that Incentive and Sender-Driven do not perform well

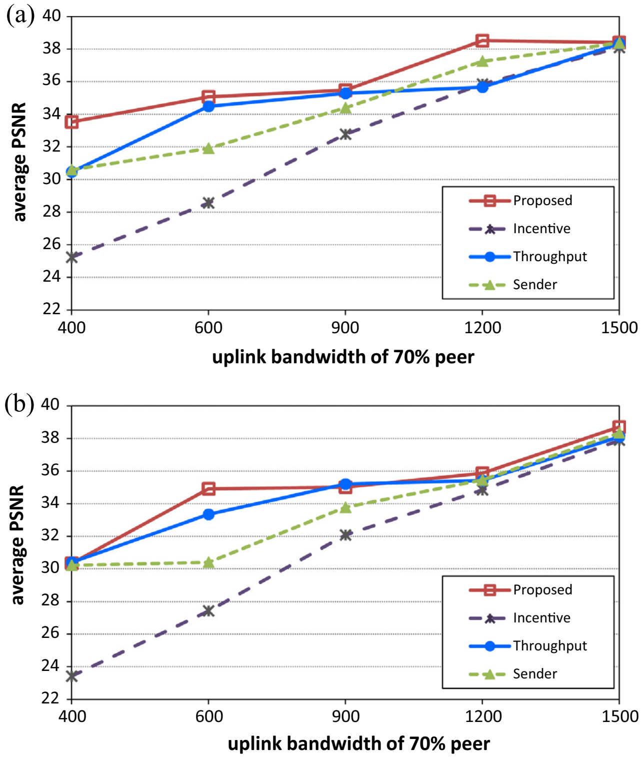

Fig. 6. PSNR performance (Foreman sequence) for peers with various uplink bandwidths under two different link packet drop rate ranges: (a) [1%, 2.5%] and (b) [1%, 10%].

when the uplink bandwidth is lower than 1500 kbps. The proposed scheme outperforms Throughput significantly when the uplink bandwidth is set to 400 kbps and 1200 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 determines the numbers of source substreams and channel substreams to subscribe to 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. All schemes can obtain reliable visual quality under 1500 kbps peer uplink bandwidth which is sufficient to transmit all substreams. However, as shown in Fig. 6(b), the PSNR performances of all schemes degrade as the link packet drop rate increases to [1%, 10%]. Nevertheless, the proposed method stably achieves the best visual quality at all bit-rates under the two different ranges of link packet drop rate.

Fig. 7(a) compares the PSNR performances of four P2P streaming schemes for peers with an uplink bandwidth of 1200 kbps and heterogeneous link packet loss conditions. The results are consistent with those in Fig. 6 in which the proposed scheme outperforms the other schemes under the link packet drop rate ranges from [1%, 5%] to [1%, 10%] and is comparable with the others under the ranges from [1%, 12.5%] to [1%, 15%]. The average numbers of received substreams in the ranges [1%, 5%] and [1%, 15%] are also compared in Fig. 7(b) and (c), respectively. With the Throughput scheme, the average numbers of received substreams in layers 1 and 2 are 7.98 and 7.83, which are significantly larger than the

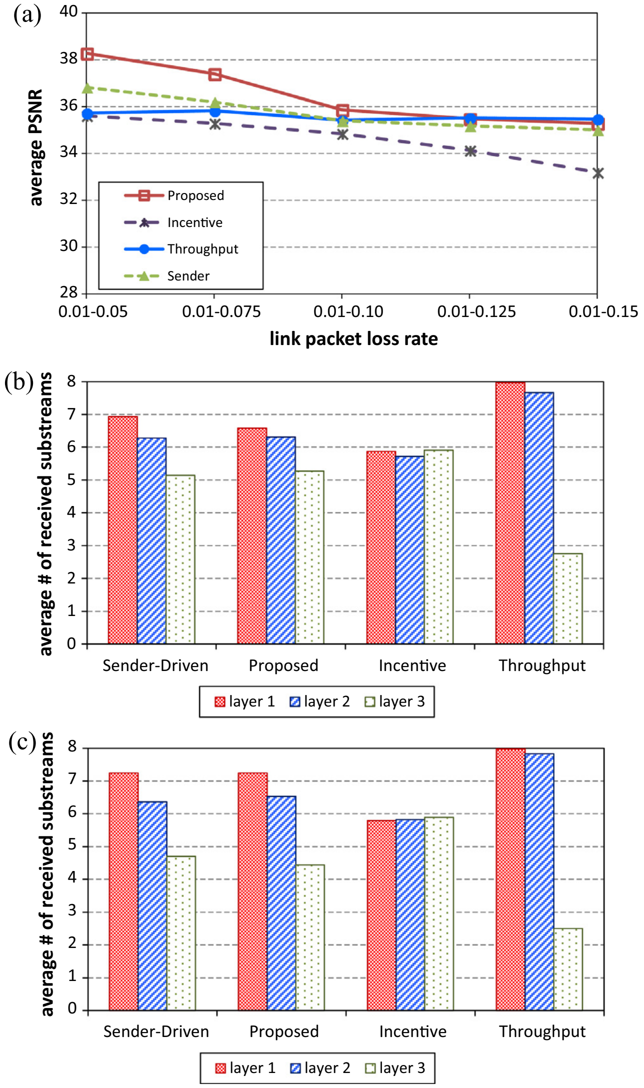

Fig. 7. Performance comparison (Foreman sequence) of four P2P streaming schemes for 70% peer with uplink bandwidth of 1200 kbps under heterogeneous link conditions: (a) the PSNR performance; (b) the average number of received substreams under link packet drop rate [1%, 5%]; (c) the average number of received substreams link packet drop rate [1%, 15%].

number 2.5 of layer 3. As a result, this protection level set still can provide reliable streaming services under a heavy packet loss condition [1%, 15%]. Under a low packet loss range of [1%, 5%], since the Throughput scheme does not consider the effect of packet loss, the average numbers of received substreams are close to that under the heavy packet loss condition. Because the uplink bandwidth is largely occupied to transmit the excessive redundant substreams in lower layers, the number of received substreams of layer 3 is not sufficient to increase visual quality. As a result, the increase of PSNR performance is insignificant under the low packet drop rate ranges. In the Incentive scheme, because the substreams in regular subscriptions are not prioritized among different layers, the lower layers (i.e., layers 1 and 2) are not well protected. Moreover, parent-peers tend to allocate more of their uplink bandwidth to the child-peers who have also contributed more uplink bandwidth to the parent-peers. Therefore, the Incentive scheme cannot accurately reflect the rate-distortion importance of substreams. As a result, the numbers of received substreams for three layers are almost identical, thus making this protection level set not able to tolerate heavy packet loss. In the Sender-Driven scheme, the parent-peers perform the contribution-guided peer selection [5] to allocate their uplink bandwidth to the subscriptions from higher-contribution client-peers. However, these accepted subscriptions usually do not compose the optimal protection level set which leads to the minimal expected distortion. Hence, the Sender-Driven scheme cannot achieve as good PSNR performance as the proposed scheme does. With the proposed method, thanks to the accurate packet loss models, each peer can subscribe to an appropriate number of redundant substreams for each layer to receive reliable streaming quality. As depicted in Fig. 7(b), peers receive more substreams of layer 3 on average [i.e., 5.27 versus 4.23 in Fig. 7(c)] under the low packet loss condition, thereby increasing the quality gain obtained from higher SNR layers. In addition, as shown in Fig. 7(c), when the packet drop rate increases to [1%, 15%], peers receive more substreams of layers 1 and 2 (i.e., [7.27, 6.53] in Fig. 7(c) versus [6.58, 6.31] in Fig. 7(b)) to obtain the quality gain from packet loss recovery. Nevertheless, the number of received substreams of layer 3 is not sufficient to recover the packet loss under the heavy packet drop rate range [1%, 15%]. Hence, the average PSNR is not as good as that of the low packet loss case.

Fig. 8(a) compares the PSNR performances of four P2P scalable streaming schemes under different mean user viewing time settings. The average numbers of received substreams for user viewing time 30 min and 5 min are also compared in Fig. 8(b) and (c), respectively. When a parent-peer unexpectedly leaves a P2P streaming system, its descendants would suffer from burst packet loss, thereby leading to severe degradation of decoded video quality if the video packets are not well protected. Since the Incentive scheme does not take into account the peer departure behavior, its performance is worse than the other three schemes. In the Throughput scheme, the numbers of received layered substreams are very close under all viewing time settings, as depicted in Fig. 8(b) and (c). Compared with the case of short user viewing time of 5 min, when the user viewing time increases to 30 min, the video quality cannot be further improved because the number of received layer-2 substreams does not increase. In contrast, since the packet loss model in the proposed scheme takes into account peer dynamics behaviors, it can maximize the quality gain of subscribed substreams under different levels of peer dynamics. As depicted in Fig. 8(b), on average peers receive more layer-2 substreams [6.01 versus 4.87 in Fig. 8(c)] under low peer dynamics. As shown in Fig. 8(c), peers receive more layer-1 substreams [6.36 versus 5.87 in Fig. 8(b)] under heavy peer dynamics. Although the number of received layer-1 substreams with the proposed scheme is fewer than that with the Throughput scheme, both numbers are sufficient to fairly well recover the packet loss caused by the heavy peer dynamics. Furthermore, since the more layer-2 substreams received with the proposed method contributes additional quality gain, the proposed scheme outperforms the Throughput scheme under heavy peer dynamics (i.e., short viewing time). In the Sender-driven scheme [5], peers optimally allocate their uplink bandwidth to the subscriptions according to their quality gain which takes into account peer dynamics as well. Therefore, the numbers of received substreams of three layers can adapt to the different levels of peer dynamics. However, the numbers of received substreams determined by parent-peers distributedly are not optimized for child-peers. With the same number of received substreams, the proposed JSCC scheme based on the receiver-driven subscription can minimize the expected visual distortion for child-peers, thereby outperforming the others in terms of PSNR performance.

Table 4 compares the average PSNR performances of four P2P streaming schemes for transmitting four videos with different resolutions and content characteristics. The results show that in most cases the proposed scheme achieve the best visual quality (see the numbers highlighted in bold-face) compared to Sender-driven, Incentive, and Throughput. The Throughput scheme achieves comparable performance with the proposed method only at low uplink bandwidths of peers and a high link packet drop rate range. The improvement of our method comes from the more accurate packet loss model and rate-distortion model used in our scheme to determine the error protection capability of each layer according to the content characteristics (i.e., in terms of compression distortion) of layers. Hence, our scheme achieves robust visual quality for various video contents under heterogeneous network conditions.

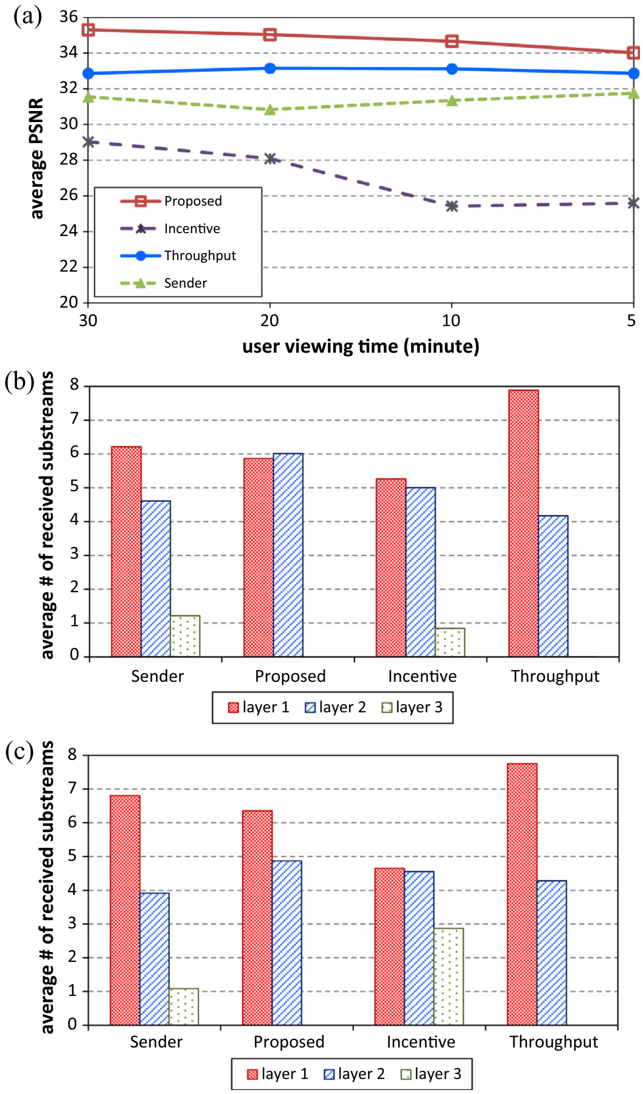

Fig. 8. Performance comparison (Foreman sequence) of four P2P streaming schemes for 70% peer with uplink bandwidth of 900 kbps under link packet loss 1%: (a) the PSNR performance; (b) the average number of received substreams under user viewing time 30 min; (c) the average number of received substreams under user viewing time 5 min.

Fig. 9 illustrates two video frames reconstructed from two layer-streams and one layer-stream., respectively, under the scenario that 70% peers have 900 kbps uplink bandwidth with 1% under link packet loss. The proposed scheme achieves the average PSNR performance of 35.07 dB, and most peers can stably receive two layer-streams, leading to the visual quality shown in Fig. 9 (a). With the other schemes, however, many peers only receive one layer-stream, resulting the visual quality as illustrated in Fig. 9(b).

Table 4
Performance comparison of four P2P streaming schemes for four video sequences.

Settings	**Proposed**		Sender-Driven [5]		Throughput [12]		Incentive [13]	
Link packet drop rate	[1%, 2.5%]	[1%, 10%]	[1%, 2.5%]	[1%, 10%]	[1%, 2.5%]	[1%, 10%]	[1%, 2.5%]	[1%, 10%]
Uplink bandwidth of 30% peers 1800 kbps	Y-PSNR (dB) of *Foreman* (CIF@30fps)							
Uplink bandwidth of 70% peers								
400 kbps	**33.52**	30.34	30.60	30.22	30.47	**30.40**	25.23	23.42
600 kbps	**35.07**	**34.91**	31.91	30.41	34.48	33.34	28.56	27.41
900 kbps	**35.48**	35.01	34.40	33.78	35.28	**35.20**	32.78	32.07
1200 kbps	**38.50**	**35.86**	37.24	35.45	35.65	35.42	35.85	34.84
1500 kbps	**38.40**	**38.40**	38.37	38.33	38.32	38.10	38.09	37.88
Uplink bandwidth of 30% peers 5000 kbps	Y-PSNR (dB) of *Football* (CIF@30fps)							
Uplink bandwidth of 70% peers								
700 kbps	**32.51**	**30.33**	29.45	29.34	31.35	29.26	30.06	28.52
1500 kbps	**33.56**	**33.17**	32.96	30.61	33.51	32.90	32.03	32.42
2300 kbps	**36.23**	**33.82**	34.17	33.34	34.04	33.65	32.48	32.22
3100 kbps	**37.82**	**37.66**	37.19	36.98	37.21	36.52	36.97	36.23
3900 kbps	**37.83**	**37.84**	37.81	37.88	37.81	37.72	37.57	37.56
Uplink bandwidth of 30% peers 10,000 kbps	Y-PSNR (dB) of *City* (4CIF@30fps)							
Uplink bandwidth of 70% peers								
2000 kbps	**31.72**	29.34	29.62	29.31	29.86	**29.36**	25.68	24.66
3600 kbps	**33.56**	**33.53**	31.85	30.31	33.18	33.14	29.97	28.63
5200 kbps	**33.76**	**33.57**	33.25	32.84	33.61	33.56	30.18	28.86
6800 kbps	**35.97**	**35.96**	35.56	35.08	34.88	34.43	34.79	33.74
8400 kbps	**36.27**	**36.21**	36.15	36.08	35.93	35.89	35.89	35.78
Uplink bandwidth of 30% peers 5000 kbps	Y-PSNR (dB) of *Ice* (4CIF@30fps)							
Uplink bandwidth of 70% peers								
1800 kbps	**37.75**	34.59	34.87	34.47	36.49	**34.69**	31.31	29.67
2600 kbps	**38.43**	**38.54**	36.74	35.18	38.30	38.09	33.13	31.29
3400 kbps	**38.69**	**38.51**	37.87	37.51	38.56	38.50	34.84	34.07
4200 kbps	**39.45**	**38.68**	39.56	38.52	38.86	38.60	37.43	37.08
5000 kbps	**40.90**	**40.77**	40.67	40.51	40.18	40.38	40.18	40.04
Uplink bandwidth of 30% peers 5000 kbps	Y-PSNR (dB) of *SaraAndKeristan* (720p@30fps)							
Uplink bandwidth of 70% peers								
1800 kbps	**38.26**	**36.32**	33.32	34.53	34.17	34.70	33.87	33.78
2600 kbps	**41.35**	**39.97**	37.74	34.16	35.40	35.41	36.27	34.57
3400 kbps	**41.47**	**41.35**	40.88	36.81	39.54	38.79	38.98	37.94
4200 kbps	41.56	**41.57**	**41.62**	40.46	40.09	40.32	40.23	40.27
5000 kbps	**41.61**	**41.59**	41.59	41.58	40.98	40.67	40.85	41.28

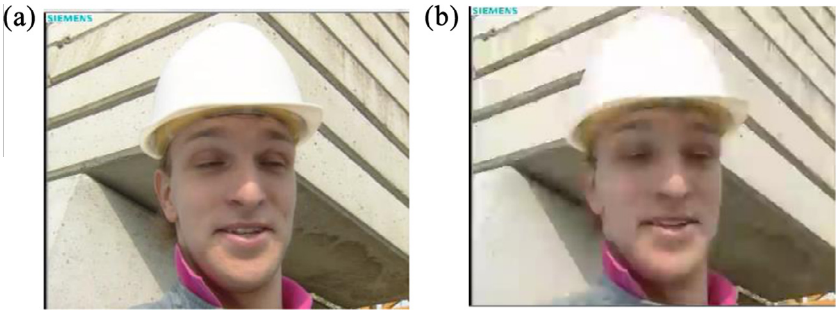

Fig. 9. Visual quality comparison (Foreman sequence) for the scenario that 70% peers have 900 kbps uplink bandwidth with 1% under link packet loss: (a) peers receive two layer-streams; (b) peers receive only one layer-stream.

5. Conclusion

In this paper, we proposed a hybrid sender/receiver-driven rate-distortion optimization framework for reliable P2P streaming. We have proposed a UEP JSCC scheme based on receiver-driven subscriptions to minimize the distortion of subscribed substreams by selecting an appropriate number of SVC and FEC redundant substreams based on the estimated global system capacity. To tackle the problem of unstable available bandwidth from parent-peers, we have proposed an uplink capacity estimation scheme based on consensus propagation to provide reliable and accurate estimates of system capacity for JSCC optimization. Furthermore, we have also proposed a sender-driven peer selection scheme which tends to reject low-contribution child-peers to maximize the rate-distortion efficiency of bandwidth allocation under the parent-peer's uplink bandwidth constraint. Our simulation results show that the proposed peer selection method achieves significantly visual quality improvement over the compared streaming methods.

References

[1] B. Akbari, H.R. Rabiee, M. Ghanbari, Packet loss in peer-to-peer video streaming over the Internet, Multimedia Syst. (2008) 345–361.

[2] C.-M. Chen, C.-W. Lin, Y.-C. Chen, Cross-layer packet retry limit adaptation for video transport over wireless LANs, IEEE Trans. Circuits Syst. Video Technol. 20 (11) (2010) 1448–1461.

[3] V. Padmanabhan, H. Wang, P. Chou, Resilient peer-to-peer streaming, in: Proc. IEEE ICNP, Atlanta, GA, 2003, pp. 16–27.

[4] P.-J. Wu, J.-N. Hwang, C.-N. Lee, C.-C. Gau, H.-H. Kap, Eliminating packet loss accumulation in peer-to-peer streaming systems, IEEE Trans. Circuits Syst. Video Technol. 19 (12) (2009) 1766–1780.

[5] C.-W. Lo, C.-W. Lin, Y.-C. Chen, J.-Y. Yu, Contribution-guided peer selection for reliable peer-to-peer video streaming over mesh networks, IEEE Trans. Circuits Syst. Video Technol. 22 (9) (2012) 1388–1401.

[6] S.-R. Tong, C. Wu, P. Phupattanasin, S.-H. Lin, Multisource FEC interleaving for mobile P2P streaming, in: Proc. IEEE Int. Conf. Multimedia Expo, San Jose, CA, 2013, pp. 1–6.

[7] H.R. Ghaeini, B. Akbari, Peer-to-peer adaptive forward error correction in live video streaming over wireless mesh network, Wired/Wireless Internet Commun., Springer-Verlag, 2014, pp. 109–121.

[8] C.-L. Chang, W.-M. Chen, C.-H. Hung, Reliable consideration of P2P-based VoD system with interleaved video frame distribution, IEEE Syst. J. 8 (1) (2014) 304–312.

[9] M. Zorzi, Performance of FEC and ARQ error control in bursty channels under delay constraints, in: Proc. IEEE Con. Vehicular Technol., 1998, pp. 1330–1394.

[10] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC Standard, IEEE Trans. Circuits Syst. Video Technol. 17 (9) (2007) 1103–1120.

[11] H.-Y. Chi, C.-W. Lin, Y.-C. Chen, C.-M. Chen, Optimal rate allocation for scalable video multicast over WiMAX, in: Proc. IEEE Int. Symp. Circuits Syst., Seattle, WA, USA, 2008.

[12] M. Zhang, Y. Xiong, Q. Zhang, L. Sun, S. Yang, Optimizing the throughput of data-driven peer-to-peer streaming, IEEE Trans. Parallel Distrib. Syst. 20 (1) (2009) 97–110.

[13] Z. Liu, Y. Shen, K.W. Ross, S.S. Panwar, Y. Wang, LayerP2P: using layered video chunks in P2P live streaming, IEEE Trans. Multimedia 11 (7) (2009) 1340–1352.

[14] H. Hu, Y. Guo, Y. Liu, Peer-to-peer streaming of layered video: efficiency, fairness and incentive, IEEE Trans. Circuits Syst. Video Technol. 21 (8) (2011) 1013–1026.

[15] X. Xiao, Y. Shi, Q. Zhang, J. Shen, Y. Gao, Toward systematical data scheduling for layered streaming in peer-to-peer networks: can we go farther?, IEEE Trans Parallel Distrib. Syst. 21 (5) (2010) 685–697.

[16] W. Ji, Z. Li, Y. Chen, Joint source-channel coding and optimization for layered video broadcasting to heterogeneous devices, IEEE Trans. Multimedia 14 (2) (2012) 443–455.

[17] C. Hellge, D. Gomez-Barquero, T. Schierl, T. Wiegand, Layer-aware forward error correction for mobile broadcast of layered media, IEEE Trans. Multimedia 13 (3) (2011) 551–562.

[18] L.X. Liu, G. Cheung, C.-N. Chuah, Rate-distortion optimized joint source/channel coding of WWAN multicast video for a cooperative peer-to-peer collective, IEEE Trans. Circuits Syst. Video Technol. 21 (1) (2011) 39–52.

[19] M. Stoufs, A. Munteanu, J. Cornelis, P. Schelkens, Scalable joint source-channel coding for the scalable extension of H.264/AVC, IEEE Trans. Circuits Syst. Video Technol. 18 (12) (2008) 1657–1670.

[20] E. Maani, A.K. Katsaggelos, Unequal error protection for robust streaming of scalable video over packet lossy networks, IEEE Trans. Circuits Syst. Video Technol. 20 (3) (2010) 407–416.

[21] C.-M. Chen, C.-W. Lin, Y.-C. Chen, Adaptive error resilience transcoding using prioritized intra-refresh for video multicast over wireless networks, Signal Process.: Image Commun. 22 (3) (2007) 277–297.

[22] S. Xie, B. Li, G.Y. Keung, X. Zhang, Coolstreaming design, theory, and practice, IEEE Trans. Multimedia 9 (8) (2007) 1661–1671.

[23] PPStream [Online]. Available: <http://www.ppstream.com>.

[24] PPLive [Online]. Available: <http://www.pplive.com>.

[25] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE 95 (1) (2007) 215–233.

[26] C.C. Moallemi, B.V. Roy, Consensus propagation, IEEE Trans. Inf. Theory 52 (11) (2006) 4753–4766.

[27] I.C. Paschalidis, B. Li, Energy optimized topologies for distributed averaging in wireless sensor networks, IEEE Trans. Autom. Control 56 (10) (2011) 2290–2304.

[28] D. Bickson, D. Malkhi, A unifying framework of rating users and data items in peer-to-peer and social networks, Peer-to-Peer Netw. Appl. J. (2008). Springer-Verlag.

[29] E. Setton, P. Baccichet, B. Girod, Peer-to-peer live multicast: a video perspective, Proc. IEEE 96 (1) (2008) 25–38.

[30] Y. He, I. Lee, L. Guan, Distributed throughput maximization in P2P VoD applications, IEEE Trans. Multimedia 11 (3) (2009) 509–522.

[31] A. Habib, J. Chuang, Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming, IEEE Trans. Multimedia 8 (3) (2006) 610–621.

[32] T. Xu, T. Zhang, Variable shortened-and-punctured Reed-Solomon codes for packet loss protection, IEEE Trans. Broadcast. 48 (3) (2002) 237–245.

[33] T. Corman, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, third ed., The MIT Press, 2009.

[34] P2Pstrsim [online]. Available: <http://media.cs.tsinghua.edu.cn/~zhangm/>.

[35] E.W. Zegura, K.L. Calvert, S. Bhattacharjee, How to model an internetwork, in: Proc. IEEE INFOCOM, vol. 2., San Francisco, CA, Mar. 1996, pp. 594–602.

[36] Z. Li, J. Chakareski, X. Niu, Y. Zhang, W. Gu, Modeling and analysis of distortion caused by Markov-model burst packet losses in video transmission, IEEE Trans. Circuits Syst. Video Technol. 19 (7) (2009) 917–931.

[37] H.264 SVC Reference Software (JSVM 9.15) and Manual CVS sever [online]. Available: Telnet: garcon.ient.rwth-aachen.de.