

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Multi-Video-Sources Selection Strategy in Mobile P2P Streaming Media Architecture

Mande Xie

College of Computer Science and Information Engineering,
Zhejiang Gongshang University, Hangzhou, Zhejiang, 310018, People's Republic of China

Abstract: A mobile peer to peer architecture for streaming media system was firstly proposed. According to the architecture, a serial scheduling and parallel scheduling algorithm were proposed for multi-video-sources. If quality of service monitored by the receiving-peer is degraded, the serial scheduling algorithm triggered the video source change event and synchronized the multi-video-sources by the time model of the streaming sequence. If the multi-video source concurrently sent the data to the receiving-peer, the parallel scheduling algorithm assigned the transmission task by frame-level bit assignment strategy. The algorithm firstly formulated the assignment problem to a nonlinearity programming problem and then transformed it into an integer programming problem based on the piece linear rate-distortion model. At last, the assignment problem was transformed into a general linear programming problem by relaxing the constraint condition. The algorithm was implemented by a greedy strategy and a piece approximate method was designed to map the solution between a linear programming problem and an integer linear programming problem. The experiment results show the algorithm is accuracy and valid and the distortion brought by the approximate method is neglected. The algorithm is very suited to a mobile P2P video streaming system.

Key words: Mobile P2P, integer linear programming, fine granular scalable code, piece linear R-D model

INTRODUCTION

With the rapid improvement of the performance of mobile telephone, people have already looked on it not only as a personal information processing platform but also as an important entertainment platform. When they are in the journey or idle, people have been accustomed to get and watch the interested video by the mobile telephone or the other mobile device. In P2P (Peer to Peer) network, the communication between the peers need not depend on a DNS(Domain Name Server) and a server and peers can directly exchange data each other. The distributed architecture makes the P2P system robust and the characteristic of P2P network makes it be suitable to the mobile network applications. As the mobile techniques and P2P network rapidly develop, the wide application prospect of mobile P2P streaming media system can be predicted.

In mobile P2P streaming media system, because the upload bandwidths of the service peers were limited, the receiver peers often needed to request the data from multi-video-sources. Nguyen and Zakhor (2002) showed the feasibility and benefits of streaming from multiple servers to a single receiver in the distributed video streaming framework. At the same time, they proposed a rate allocation algorithm to determine the sending rate of each server to minimize the total packet loss. The rating assignment problem for multi-version streams was studied

by Liu *et al.* (2006) and they computed an optimal streaming rate for each stream version. Similar to Liu *et al.* (2006), Hsu and Hafeeda (2007a) and Magharei and Rejaie (2006) employed layered-scalable coding to adapt to the heterogeneous bandwidth. The multi-sender streaming problem was formulated to maximize the streaming quality of all peers and minimize the load on the originating media distributor (Hsu and Hafeeda, 2007b). Magharei and Rejaie (2006) designed a practical algorithm to adapt to bandwidth dynamics. However, they only considered coarse-grained scalability and are not R-D optimized. Su and Wang (2006) considered a minimization problem for the transmission of FGS-coded images. They formulated the problem as a nonlinear programming problem and transformed it to a series of Linear Programming (LP) sub-problems, where each sub-problem determined a peer allocation to maximize the number of received bits in a given time bound. However, they didn't consider the characteristic of R-D models. Hsu and Hafeeda (2007a) considered the relationship between coding efficiency of FGS and served-custom range. After, they traded off the coding efficiency of FGS and served-custom range, they proposed an algorithm to determine the best base layer bit rate of single or multiple video streams. However, their optimized object is video source end instead of video client end. The bit distribution problem at different aggregation levels for FGS-encoded sequences was studied. De Cuetos *et al.* (2005)

formulated a minimization problem for the total distortion for a block of frames of different sequences and solved it using dynamic programming method. They had validated frame aggregations reduced the complexity of dynamic programming method without compromising the quality. But the algorithm required each sender had a complete video stream. Nemati *et al.* (2008) proposed a multi-source streaming model for mobile peer-to-peer (P2P) overlay network and they scheduled the multi-source by serial way.

In mobile P2P streaming media system, there often exists multiple video sources. This study is to solve the serial and parallel scheduling problem for multiple video sources and the bit assignment problem of stream.

THE VIDEO STREAMING ARCHITECTURE BASED ON MOBILE P2P NETWORK

An architecture of a mobile P2P network employs the framework of the super-unstructured P2P architecture. The framework divides all peers into two or several layers by their comprehensive capacity. The peers with strong power become super-peers and the other peers become leaf peers. Each leaf peer is connected to an adjacent super-peer and each super-peer connects some leaf peers and some other super-peers. The information of shared

files is only saved in super-peers. This architecture integrates several merits, such as high efficiency of centralized search, autonomy, load balance and robustness of decentralized search. At the same time, it trades off the centralized structure and the decentralized structure and makes full use of the difference of peers' capacity. However, unlike to the cable network, the mobile P2P architecture cannot directly employ the present P2P architecture. The operation of a mobile network is controlled by its operation business and the network model for P2P application must be controllable. Hence, based on the super unstructured P2P network, all peers in the architecture are factitiously classified into two categories: General Super-Peers (GSP) and the Special Super-Peers (SSP) controlled by the mobile-network operator. A general fixed peer is connected to a GSP and a mobile device is connected to SSP. Consequently, the original super-unstructured P2P architecture need almost not be changed and only some peers controlled by the mobile-network operator join the system and become super-peers. The original fixed peers are not affected. Figure 1 shows the mobile P2P architecture.

Because the present mobile network employs layered structure, pure P2P networks don't exist and the datum can not be directly shared between peers. Hence, some independent mobile agent servers are introduced and all

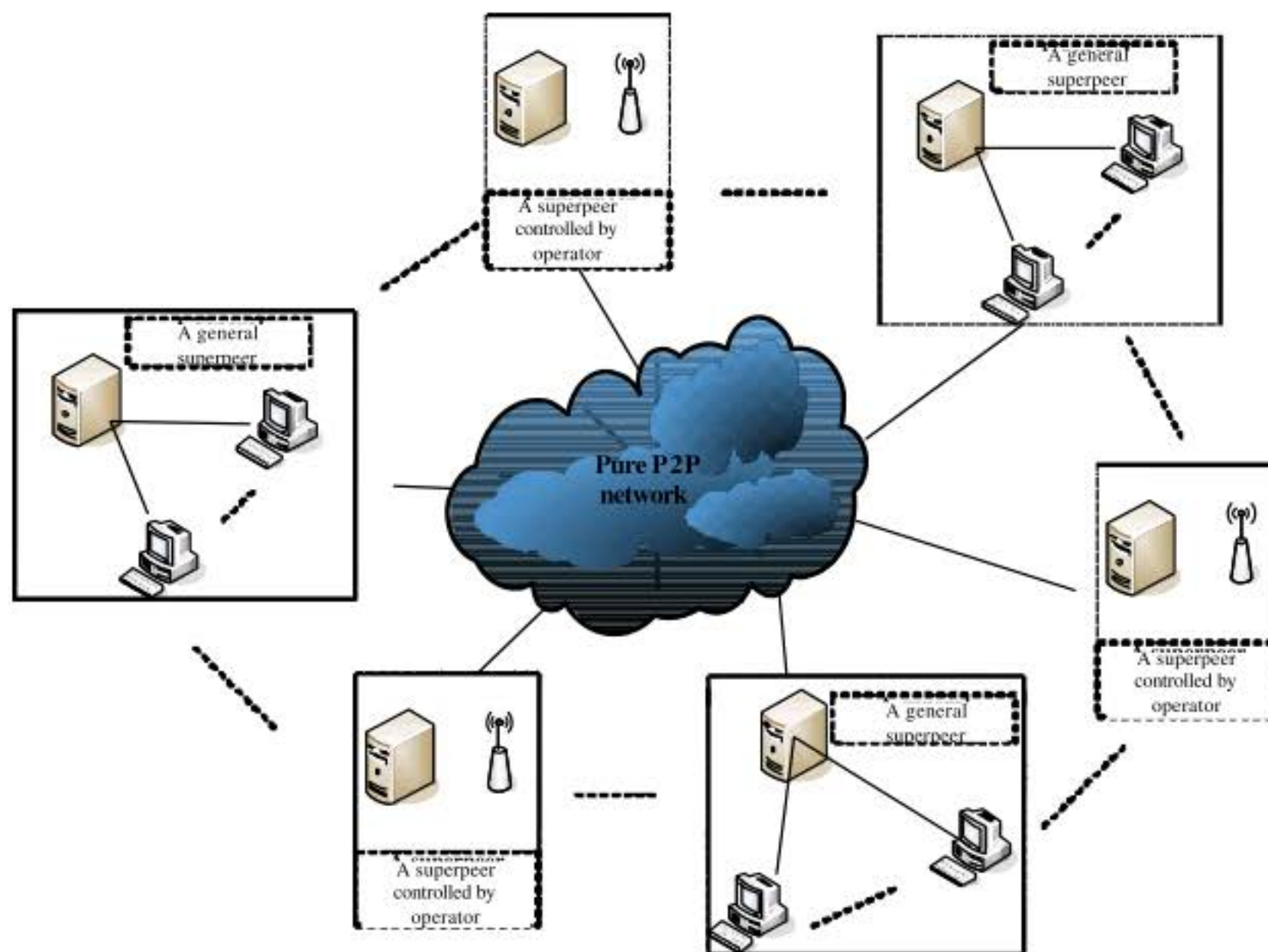


Fig. 1: The general mobile P2P architecture

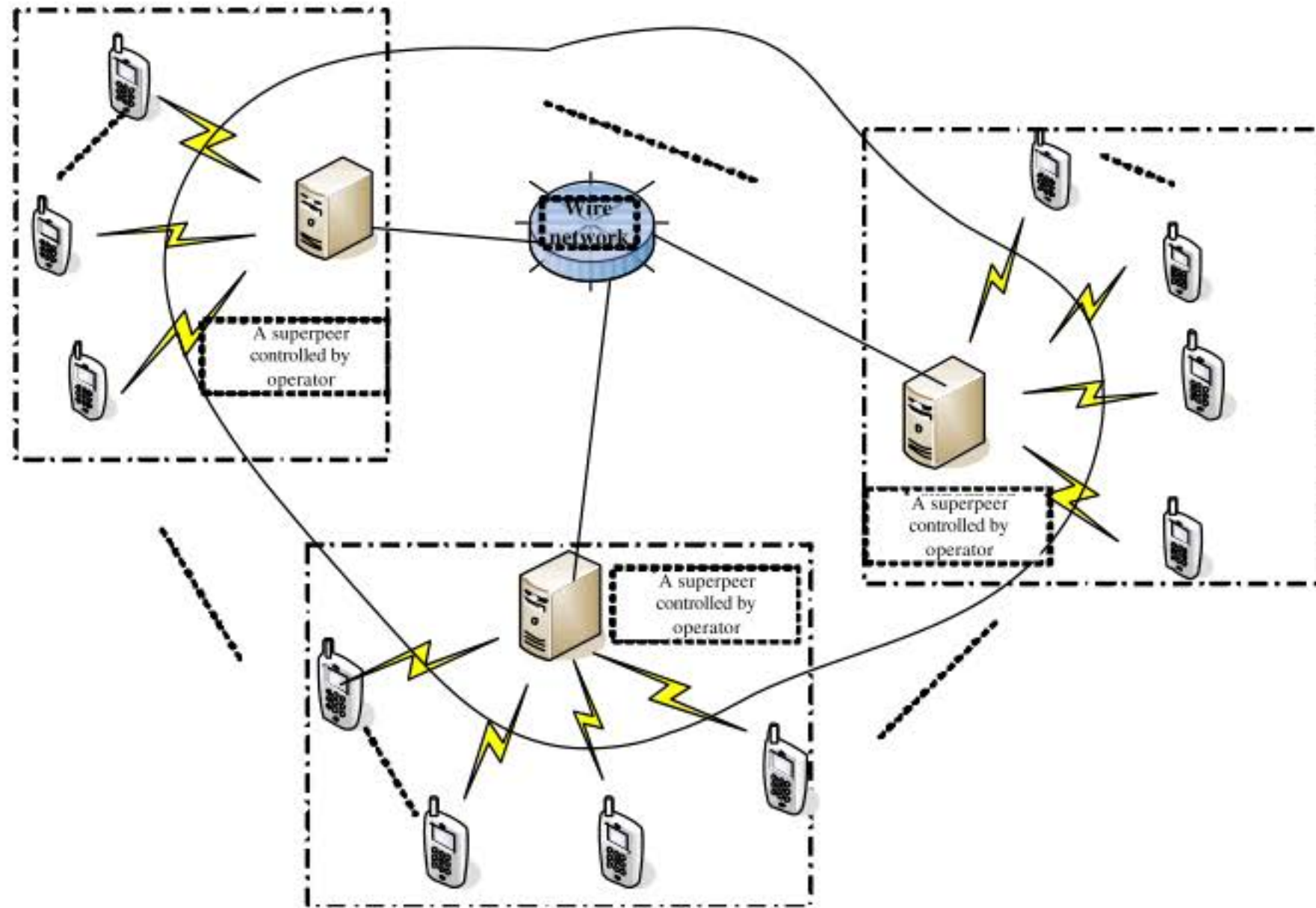


Fig. 2: The mobile P2P architecture with agent servers connecting some mobile devices

mobile devices finish dual connection by it. Figure 2 shows the architecture with agent servers. In this architecture, a group of mobile devices are connected to an agent server and the identifier of each mobile node is announced in the mobile P2P network such that each mobile device can independently work by an agent server. In fact, a super-peer controlled by mobile-network operator can be viewed as an independent mobile P2P agent server, which can capture all functions of an agent sever. In the mobile P2P video streaming system, the bandwidth and operation power of mobile devices are limited. Hence, in the architecture, the multiple video sources must be correctly scheduled in order to improve the bandwidth utilization ratio of the receiving peer, video perceived quality and meet the real-time requirement.

THE FRAMEWORK OF THE ALGORITHM

Figure 3 shows the framework of this algorithm. A peer firstly submits the streaming media service search request to the connected superpeers and superpeers find the service peer set which satisfies the expected Qos by the common resource search algorithms. If the service peer set isn't empty, the multiple video sources are scheduled by a serial strategy and the different video

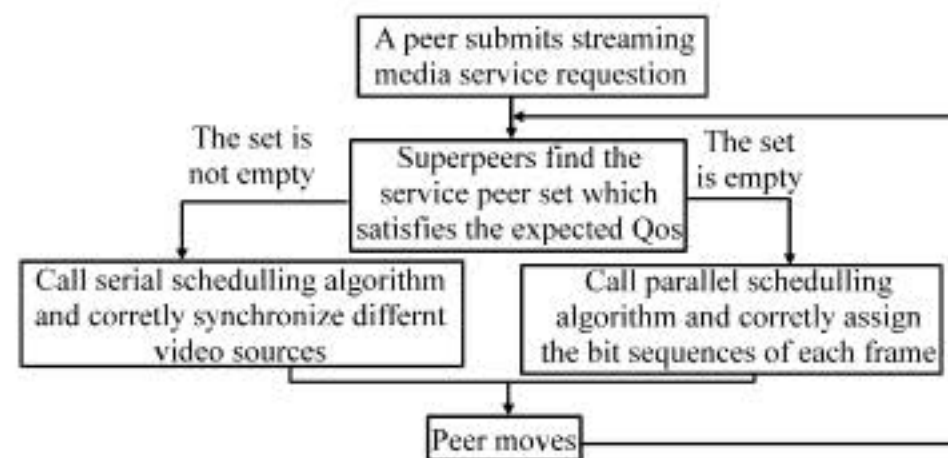


Fig. 3: The framework of this algorithm

sources are synchronized by the time model of a streaming. If the service peer set is empty, it is proved that there no any service peer can alone provide the expected Qos. The superpeers start to find the service peer set which possesses the requested media content and the expected Qos is satisfied by parallel way in which multiple video sources concurrently serve a receiving peer. In parallel method, the frame bit assignment algorithm is presented to maximize the perceived quality. If the receiving peer has moved, it just reconnects a new superpeer and repeats the above process. Next, the serial and parallel scheduling methods are described in detail.

THE SERIAL SCHEDULING ALGORITHM FOR MULTIPLE VIDEO SOURCES

If the service peer set which satisfies the expected Qos isn't empty, let SP indicate the service peer set and let P_r indicate the current requesting peer. Because SP is not empty, there are some service peers which each of can provide the expected Qos and multiple vide sources are scheduled by serial way. Firstly, P_r randomly chooses a service peer from this set to request stream data and monitors the Qos provided by this service peer in real time. If the Qos is degraded for the moving of this service peers, communication congestion or declining in service capacity, P_r re-chooses a service peer from this set to request stream data. Because a different service peer may have a different stream version, such as a different resolution and a frame rate, multiple video sources must be synchronized to play seamlessly video in client end when a service peer is switched.

The synchronization method between multiple video sources: In fact, a video streaming just is a bit sequence. If two service peers have the same streaming version, the same bit sequence is transmitted and vice versa. As a result, a service peer is switched to another service peer which possesses the different stream version, the multiple video sources must be synchronized. Let BS_i indicate the bit sequence sent by the service peer SP_i , let $|BS_i|$ indicate the length of BS_i , let BT_i indicate the time length to play BS_i and let $BS_i^{t_1} t_2$ indicate the sub-sequence of BS_i which is between the time t_1 and the time t_2 . Obviously, the average bit rate $Br_i = |BS_i|/BT_i$. Suppose the service peer P_i starts to send the stream data to P_r at time t_0 and another service peer P_j takes over the service peer P_i at time t_1 because the Qos of P_i is degraded and the subsequence sent by P_i is $BS_i^{t_0} t_1$. For the same time, another service peer P_k takes over the service peer P_j at time t_2 and the subsequence sent by P_j can be marked as $BS_j^{t_1} t_2$ and so on. Figure 4 shows the process. Obviously, when P_i , P_j and P_k have different stream version, the time variable t records the length of the sent subsequence and multiple video sources can be effectively synchronized by it. When a service peer is switched, one just passes the time parameter t to the new service peer and let the new service begin to send the stream data at time t . At the same time, in order to ensure that P_r can play seamlessly the received video, let each service peer transmit some margins of stream data to consider the synchronization delay. As shown in Fig. 4, the length of time in which P_i sends the stream data is δ longer than t_1-t_0 and the start time at which P_j starts to send the stream data is δ earlier than the time t_1 .

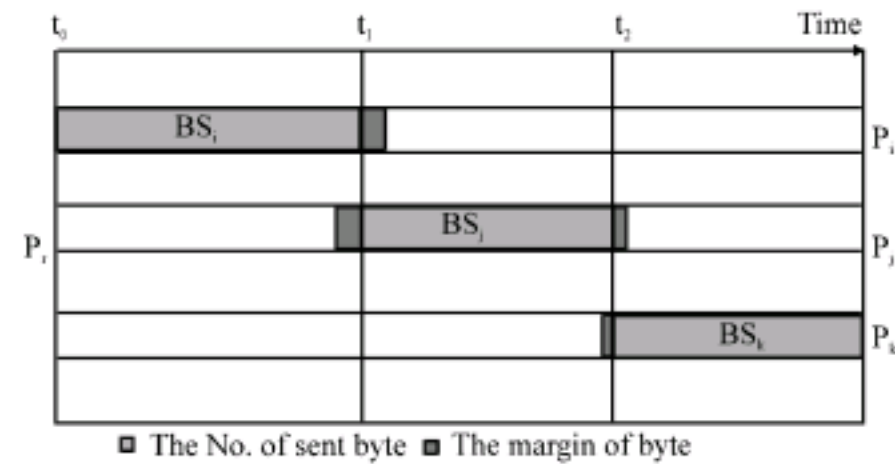


Fig. 4: The synchronization theory of multiple video sources

THE PARALLEL SCHEDULING ALGORITHM FOR MULTIPLE VIDEO SOURCES

If the service peer set is empty, it is proved that there no any service peer can alone provide the expected Qos. The superpeers start to find the service peer set which possesses the required media content and the expected Qos is satisfied by parallel way in which multiple video sources concurrently serve a receiving peer. Because each video source has a different part of a sequence and different upload bandwidth, the receiving peer must properly choose video sources and reasonably allocation limited download bandwidth to maximize perceived quality. This algorithm assumes the bit rate of base layer of FGS-encoded video sequence is low and can be reliably transmitted to the receiving peer. The parallel scheduling algorithm mainly optimizes and reasonably allocates the bit stream of the enhancement layers of FGS-encoded video sequence, because its bit rate is high and controllable. Because a video sequence is composed of lots of frames, a parallel scheduling problem for multiple video sources can be transformed into receiving-peer-driven frame-level bit allocation problem. In another word, a parallel scheduling algorithm is to allocation the transmission task of each frame to each video source such that the perceived quality of a receiving peer is maximized.

The model of the bit allocation problem: For ease of exposition, let T indicate a frame period, which is fixed in a sequence; let n indicate the number of video sources; let $bw_i (i \in [1, n])$ indicate the bandwidth of the video sources i (these information can be collected by peer exchange or the other protocols); let e_i indicate the number of successive bits of the enhancement layer possessed by video source i and without loss of generality, let $e_1 \leq e_2 \leq \dots \leq e_n$; let R indicate the bandwidth of receiver; let Δ_i indicate the number of bits which are allocated to video source i and let r_i indicate the bit rates which are allocated to video source i . And, then the optimized problem can be formalized into selecting the optimal $\Delta_i (i \in [1, n])$ and

$r_i(i \in [1, n])$ to maximize the perceived quality. If the perceived quality is represented by the distortion and $D(r)$ is denoted as the R-D function, then the bit allocation problem can be modeled as:

$$\begin{aligned} \min_{\Delta} D(\sum_{i=1}^n \Delta_i) \quad (1) \\ \text{s.t. } \Delta_i - r_i T \leq 0, \sum_{i=1}^n \Delta_i \leq e_i, r_i \leq bw_i \\ \sum_{i=1}^n r_i \leq R \end{aligned}$$

and $\Delta_i, r_i \in \mathbb{N}; i = 1, 2, \dots, n$, where, $\Delta_i - r_i T \leq 0$ ensures that the number of bits which are allocated to the video source i is less than its transmission capacity;

$$\sum_{i=1}^n \Delta_i \leq e_i$$

ensures that the number of bits which are allocated to the video source i is within the portion of bits saved by the video source i ; $r_i \leq bw_i$ ensures that the bit rate which is allocated to the video source i is less than its upload bandwidth and

$$\sum_{i=1}^n r_i \leq R$$

ensures that the sum of received bit rate is less than its download bandwidth. As shown in Eq. 1, this problem is a non-linear programming problem (NLP) which is a NP-complete. Assume Δ_i comes from j different bit-planes and seg_{it} ($t \in [1, j]$) represents the number of bits of the bit plane t and then

$$\Delta_i = \sum_{t=1}^j seg_{it}$$

where, j is a variable. Therefore, the Eq. 1 can be transformed into the Eq. 2.

$$\min_{\Delta} D(\sum_{i=1}^n \Delta_i) = \min_{\Delta} D(\sum_{i=1}^n \sum_{t=1}^j seg_{it}) \quad (2)$$

Equation. 1 and 2 indicate the choice and design of R-D function $D(r)$ are very important to simplify the problem. Hsu and Hefeeda (2008) has shown the piece linear R-D model is accurate and low time complexity. What is more important is the property of the piece linear R-D model is helpful to the design of the optimizing algorithm. Hence, the algorithm employs the piece linear R-D model as $D(r)$. Based on the piece linear R-D model

and unwrapped Eq. 2, all items which belong to the same bit-plane are merged into a new item denoted as $bp_v(v \in [1, z])$, where, bp_v is the number of transmitted bits from the bit-plane v and z is the total number of bit-planes of the considered frame. Let g_v is the slope of the bit-plane v and then the bit allocation problem can be further converted to:

$$\begin{aligned} \min_{\Delta} D(\sum_{i=1}^n \Delta_i) = \min_{\Delta} D(\sum_{i=1}^n \sum_{t=1}^j seg_{it}) \quad (3) \\ = d + \sum_{v=1}^z g_v \cdot bp_v \end{aligned}$$

$$\text{s.t. } \sum_{i=1}^n \Delta_i \leq e_i, r_i \leq bw_i, \sum_{i=1}^n r_i \leq R, bp_v \leq l_v, \sum_{i=1}^n \Delta_i = \sum_{v=1}^z bp_v$$

and $\Delta_i, r_i, bp_v \in \mathbb{N}; i = 1, 2, \dots, n; v = 1, 2, \dots, z$, where, $l_v(v = 1, 2, \dots, z)$ is the size of the bit plane v ; d is the distortion when only the base layer is transmitted; the new constraint condition $bp_v \leq l_v$ ensures the number of allocated bits is less than the size of the bitplane v ;

$$\sum_{i=1}^n \Delta_i = \sum_{v=1}^z bp_v$$

ensures that the total number of bits transmitted from different bitplanes is exactly the same as the number of bits allocated to the video sources. As shown in the Eq. 3, the original NLP problem is transformed into an Integer Linear Programming (ILP) problem. However, it is still a NP-complete. $\Delta_i, r_i, bp_v \in \mathbb{N}; i = 1, 2, \dots, n; v = 1, 2, \dots, z$ is relaxed to $\Delta_i, r_i, bp_v \in \mathbb{R}; i = 1, 2, \dots, n; v = 1, 2, \dots, z$, such that the ILP problem can be transformed into a general Linear Programming (LP) problem, which is easily solved with the simplex method or the other existing method.

The solutions of the LP problem and the (ILP) problem are approximately mapped by a simple method.

Assume that:

$$A^* = \{(\Delta_i^*, r_i^*) | i = 1, 2, \dots, n\}$$

is the optimal solution of LP problem, where, Δ_i^*, r_i^* indicate some non-negative real number and then the solution of ILP can be gotten by the followed piecewise function.

$$\begin{aligned} \bar{r}_i = \lfloor r_i^* \rfloor \\ \bar{\Delta}_i = \begin{cases} \lfloor \Delta_i^* \rfloor & \frac{\Delta_i^*}{r_i^*} \leq T \\ \lfloor \Delta_i^* \frac{r_i^* - 1}{r_i^*} \rfloor & r_i^* > 1 \text{ and } \frac{\Delta_i^*}{r_i^*} \leq T \\ 0, & 0 \leq r_i \leq 1 \text{ and } \frac{\Delta_i^*}{r_i^*} \leq T \end{cases} \quad (4) \end{aligned}$$

Obviously, all constrain conditions except for $\Delta_i - r_i T \leq 0$ can be naturally satisfied and the $\Delta_i - r_i T \leq 0$ is ensured by the piecewise function 4. Where, r_i^* is rounded down and the maximum error is not more than one. For Δ_i^* , when

$$\frac{\Delta_i^*}{r_i^*} \leq T$$

it is directly rounded to an integer and the maximum error is not more than one. As a result, the cumulative error is not more than n and the cumulative error is not more than $nT+n$ in other two cases. Obviously, the approximate solution of ILP gotten by the piecewise function is close to his optimal solution. A greedy strategy is employed in the design and implement on the ILP to accelerate the algorithm. The basic idea of greedy strategy is to maximize the number of bytes sent by each video source and to make full use of the download bandwidth of the receiving peer.

THE EXPERIMENT RESULTS

All algorithms proposed in this study have been implemented and validated. Because the serial scheduling

algorithm for multiple video sources is relatively simple, the experiment results shown here are only for the parallel scheduling algorithm. From lots of video sequences, four typical test sequences are chose such as coastguard, flower, foreman and mobile. Several experiment cases are also constructed and the Table 1 summarizes these cases. In Table 1, the incoming bandwidth of 1, 2, 4 and 100 M simulates the present popular internet and intranet streaming setting. The third column in the Table 1 enumerates the setting of all video sources and in the numeric pairs, the former is outgoing bandwidth of the video source and the latter is the bit stream version.

Figure 5 and 6 show the experiment results. In figures, the vertical axis is the mean quality which is represented by PSNR (dB) and the horizontal axis is the number of frames. Figure 5a-d show the mean quality of three algorithms for Coastguard sequence in all cases, where nonscalable is nonscalable coding algorithm, Simplex is a simplex method and BitAlloc is the greedy algorithm. As shown in Fig. 6a-d, the video quality of nonscalabe is the worst and the video quality of Simlex and Bit-Alloc is very similar. Compared to Fig. 5 and 6 shows the corresponding quality improvement. The least quality

Table 1: Parameters of the four experiment cases

Case No.	The incoming bandwidth of receiver Peer (unit: bps)	The senders (outgoing bw, bitstream version)
I	1 M	(128k,128k),(128k,128k),(512k,512k),(1M,1M),(1M,1M),(1M,1M)
II	100 M (unlimited)	(256k,512k),(512k, 4M),(1M,8M),(2M,10M)
III	2 M	(512k,512k),(256k,1.5M),(512k,1.5M),(256k,1.5M),(128k,1.5M)
IV	4 M	(1M,1M), (1M,1M),(1M,1M), (1M, 512k),(512k,512k), (512k, 2M), (2M, 4M)

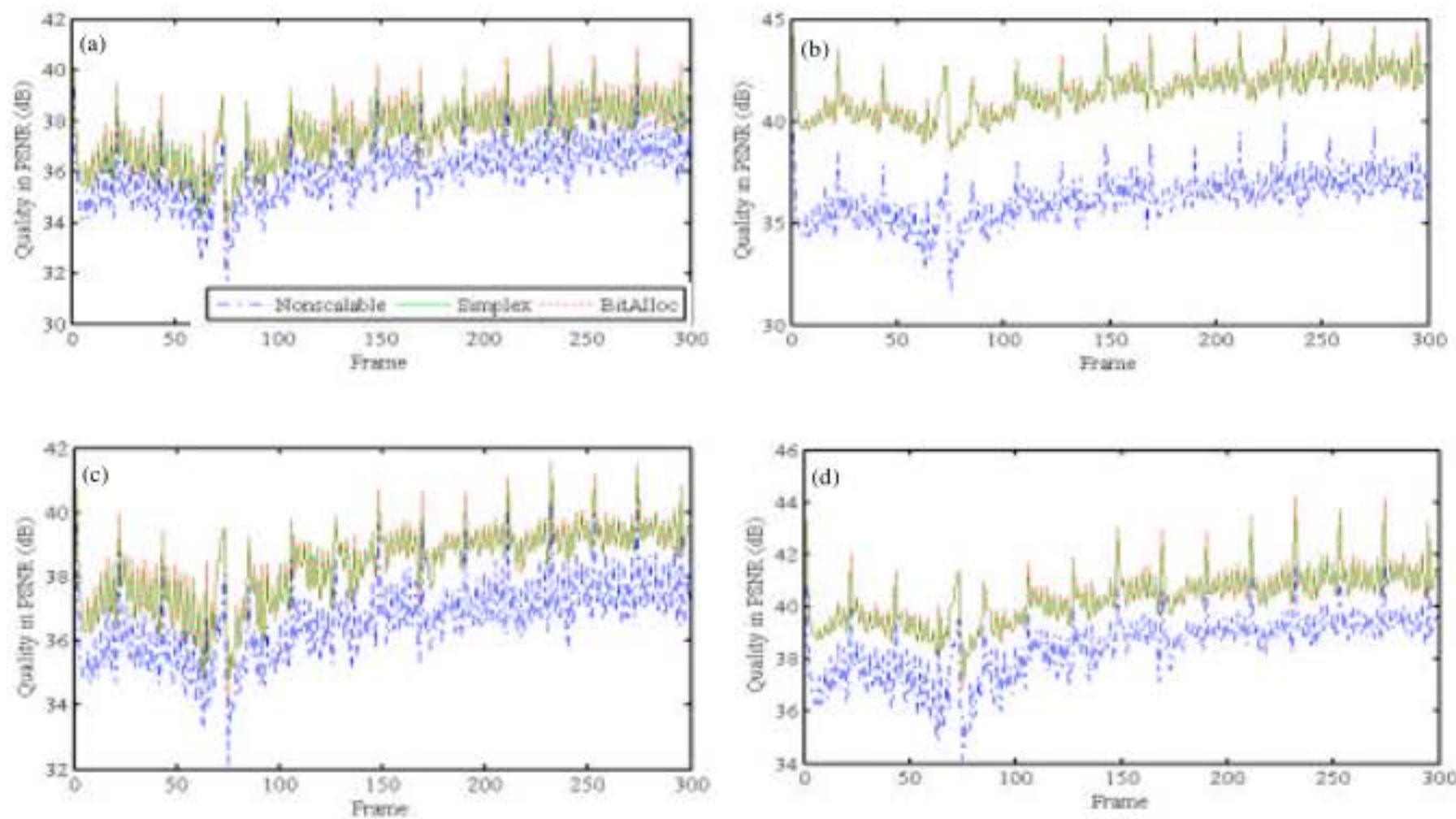


Fig. 5: The mean quality of three algorithms for (a) Case 1, (b) Case 2, (c) Case 3 and (d) Case 4 of coastguard sequence

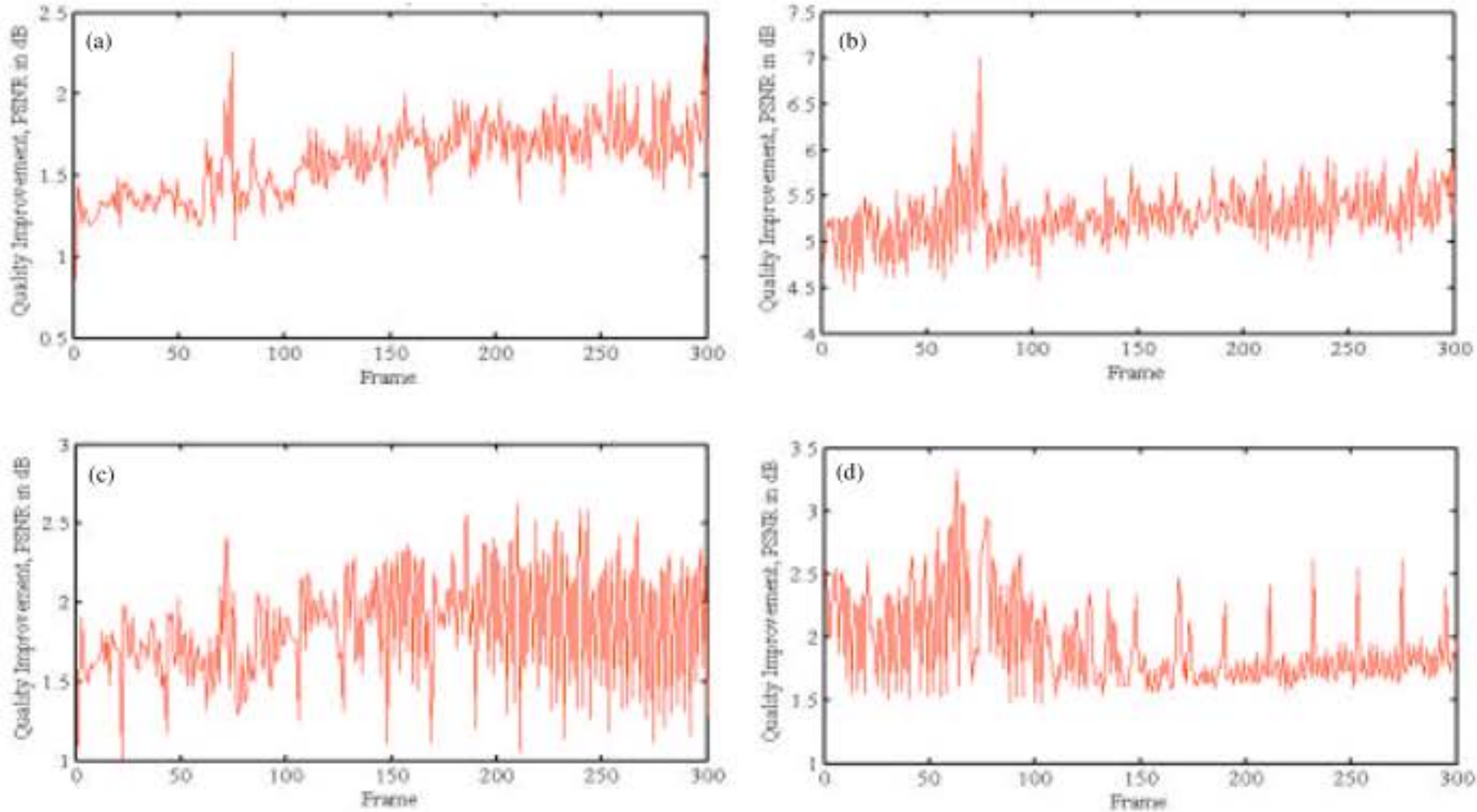


Fig. 6: The Stat of quality improvement of the coastguard sequence; (a) Case 1, (b) Case 2, (c) Case 3 and (d) Case 4 of costguard sequence

Table 2: The stat of mean quality improvement between BitAlloc and Nonscalable (unit:dB)

Sequences	Case I	Case II	Case III	Case IV
Coastguard	1.5897	5.2845	1.8470	1.9380
Flower	2.0451	6.9780	2.1633	2.8158
Mobile	1.3039	4.6802	1.2384	1.9307
Foreman	1.5170	4.9955	1.4819	2.0337

Table 3: The mean run time in all cases (unit:ms)

Sequences	Simplex				BitAlloc			
	Case I	Case II	Case III	Case IV	Case I	Case II	Case III	Case IV
Coastguard	7.4704	5.7723	6.3517	7.1390	0.002803	0.0025534	0.0027108	0.0031457
Flower	6.9196	6.2865	6.3167	6.7486	0.002762	0.0021548	0.0022088	0.0025189
Mobile	8.9592	7.4171	7.9457	8.2259	0.0028542	0.0025245	0.0027033	0.0035051
Foreman	7.1809	5.4704	6.1104	6.8058	0.0028719	0.0025767	0.0027024	0.0029715

improvement between Simplex, BitAlloc and Nonscalable isn't less than 1dB and the most one is more than 7 dB. Table 2 shows the more comprehensive mean quality comparison. The similar results are gotten and in all cases, the least quality improvement between BitAlloc and the Nonscalable isn't less than 1.2 dB and the most one is more than 5.2 dB. Table 3 shows the comparison result in run time between Simplex and BitAlloc and as shown in Table 3, BitAlloc is efficient and the run time is 1/3000 times less than that of Simplex.

CONCLUSION

According to the architecture of streaming media system, a serial scheduling and a parallel scheduling algorithm are proposed for multi-video-sources. In serial scheduling algorithm, only a service peer transmits

the streaming data in the same moment and the multiple video sources work in turn. The theory of serial scheduling algorithm is relatively simple and is easy to understand without the experiment results. The scheduling problem of multiple video sources is relatively complex. It is modeled as a NLP and is converted to a GLP. In the implementation, the greedy strategy is introduced. The experiment results show the video quality of the parallel scheduling called BitAlloc and Simplex is similar and is much better than that of NonScalable. The reason is the bandwidth of receiving peer is made full use in BitAlloc and Simplex. However, the BitAlloc is much shorter than the Simplex in run time because the greedy strategy is introduced in the implementation of BitAlloc. The algorithms are very suited to the mobile P2P video streaming system.

ACKNOWLEDGMENTS

This study is supported in part by National Natural Science Foundation of China under Grant No.60673179 and Natural Science Foundation of Zhejiang Province under Grant No. Z106727.

REFERENCES

- De Cuetos, P., P. Seeling, M. Reisslein and K. Ross, 2005. Comparing the streaming of FGS encoded video at different aggregation levels: Frame, GoP and scene. *Int. J. Commun. Syst.*, 18: 449-464.
- Hsu, C.H. and M. Hefeeda, 2007a. Optimal partitioning of fine-grained scalable video streams. Proceedings of the of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video, Jun. 2007, Urbana-Champaign, IL, pp: 63-68.
- Hsu, C.H. and M. Hefeeda, 2007b. Structuring multi-layer scalable streams to maximize client-perceived quality. Proceedings of the 15th IEEE International Workshop on Quality of Service, Jun. 21-22, Evanston, IL, pp: 182-187.
- Hsu, C.H. and M. Hefeeda, 2008. On the accuracy and complexity of rate-distortion models for FGS-encoded video sequences. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4: 1-22.
- Liu, J.C., B. Li and Y.Q. Zhang, 2006. Optimal stream replication for video simulcasting. *IEEE Trans. Multimedia*, 8: 162-169.
- Magharei, N. and R. Rejaie, 2006. Adaptive receiver-driven streaming from multiple senders. *Multimedia Syst. J.*, 11: 550-567.
- Nemati, A.G., T. Enokido and M. Takizawa, 2008. A multi-source streaming model for mobile peer-to-peer (P2P) overlay networks. Proceedings of the 28th International Conference on Distributed Computing Systems Workshops, Jun. 17-20, Beijing, pp: 18-23.
- Nguyen, T. and A. Zakhor, 2002. Distributed video streaming over internet. Proceedings of the Multimedia Computing and Networking, 2002, San Jose, CA, pp: 186-195.
- Su, X. and T. Wang, 2006. Sequence of linear programming for transmission of fine-scalable coded content in bandwidth-limited environments. *Multimedia Syst.*, 11: 455-466.