DISTRIBUTED THROUGHPUT MAXIMIZATION IN HYBRID-FORWARDING P2P VOD APPLICATIONS

Yifeng He, Ivan Lee and Ling Guan

Department of Electrical and Computer Engineering,
Ryerson University, Toronto, Canada

ABSTRACT

In peer-to-peer (P2P) video-on-demand (VoD) systems, a scalable source coding is a promising solution to provide heterogeneous peers with different video quality. In a scalable P2P VoD system, the received video quality at each peer is dependent on the throughput if the packet redundancy has been addressed with coding techniques. In this paper, we propose a hybrid-forwarding P2P VoD architecture, which integrates both the buffer-forwarding approach and storage-forwarding approach. Furthermore, we develop a distributed algorithm to maximize the throughput. Through simulations, we demonstrate that the hybrid-forwarding architecture can obtain a much higher throughput compared to the architecture with only buffer-forwarding links or with only storage-forwarding links.

***Index Terms*—** Peer-to-peer video-on-demand, throughput maximization, distributed optimization

1. INTRODUCTION

Recently, peer-to-peer (P2P) technology has become a good approach to provide live video broadcast [1] or VoD streaming service [2, 3] to a huge number of the users over a global area. Most existing P2P VoD systems only adopt single-layer video coding [3]. If the source rate is high, some peers with a limited or low download bandwidth may not be able to accommodate it. On the other hand, if the source rate is too low, peers with a higher download bandwidth may underutilize their download bandwidth. Therefore, a scalable source coding is a promising solution for P2P VoD systems with heterogeneous bandwidth.

In a scalable P2P VoD system, the objective is to maximize the received quality at each peer. By applying coding techniques such as network coding [4], the redundant packets can be eliminated with a very high probability. If each peer receives distinct packets, maximizing the reconstructed quality at each peer is equivalent to maximizing the throughput.

The overlay construction has a great impact on the system throughput. Many of the existing P2P VoD systems use buffer-forwarding approach [2], where the participating peers organize themselves into an overlay based on the playback progress, each peer buffering the recently received content and forwarding it to the following peers. The buffer-forwarding approach moves a significant fraction of the uploading task from the server to the peers. However, a peer only redistributes the video it is currently watching to the peers that are watching the same video, and it does not redistribute the content that it has stored in its storage. Also, the upload bandwidth of the idle peers cannot be utilized since they have no buffered content.

The system throughput can be maximized through optimal link rate allocation. However, how to optimally allocate each link rate is a challenging problem. The reasons are as follows. 1) The number of link rate variables is typically huge in a P2P system. 2) There are correlations among different link rate variables. For example, the outgoing link rate and the incoming link rate at a peer must follow the flow conservation law. 3) Each peer has only a local view, thus centralized algorithm is not suitable for P2P applications.

In this paper, we propose a hybrid-forwarding P2P VoD architecture, which greatly improves the system throughput. Furthermore, we develop a distributed algorithm to maximize the throughput by optimally allocating the link rates.

2. HYBRID-FORWARDING P2P VOD SYSTEMS

A hybrid-forwarding P2P VoD architecture integrates both the buffer-forwarding approach and the storage-forwarding approach. The peers watching the same video form an overlay with the buffer-forwarding approach. In the example shown in Fig. 1, peer 3, 7, 130, 116, 104 and 1 form video-1 buffer-forwarding overlay, and peer 134, 34, 115, 43, 22 and 1 form video-2 buffer-forwarding overlay. Each peer replicates in their storage one or multiple segments of the video it has watched before. The stored

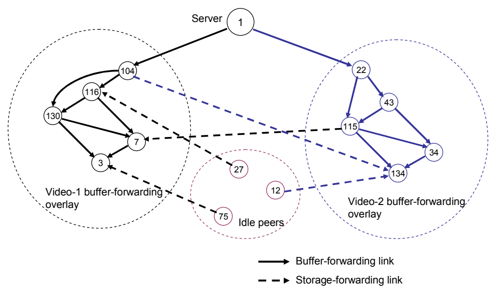

Fig. 1. A hybrid-forwarding P2P VoD system

segments can be used to serve the other peers. Some peers watching video 2, such as peer 115, can forward their stored segments to the peers watching video 1 (e.g., peer 7). The idle peers (e.g., peer 27, 12, and 75) are encouraged to contribute their stored segments to serve other peers. In the hybrid-forwarding architecture, the peers (e.g., peer 115) may have both buffer-forwarding outgoing links and storage-forwarding outgoing links. These two kinds of outgoing links participate in different video sessions, and compete for the limited upload bandwidth. In VoD applications, different video session has different priority. We can implement service differentiation among videos by allocating a larger portion of the upload bandwidth to the links participating in a high prioritized video session compared to those participating in a low prioritized video session. Furthermore, this hybrid architecture is robust to random seeks. For example, when the buffer-forwarding parents of peer 7 (e.g., peer 130 and 116) both jump to other positions, peer 7 can still have a stream supply from its storage-forwarding parent (e.g., peer 115).

In the overlay construction, the peers watching the same video are first organized into a buffer-forwarding overlay using one of the existing approaches in [2, 5], then each peer locates the storage-forwarding parents either in a centralized way [9] or in a distributed way [3] and connect itself to them. The overlay construction module handles the peer dynamics and output the overlay to the throughput maximization module.

3. DISTRIBUTED THROUGHPUT MAXIMIZATION

3.1. Problem formulation

The hybrid-forwarding mesh overlay can be represented by a directed graph $G = (N, L)$, where the set of the peers are denoted by i=1,…,N, and the set of directed overlay links are denoted by l=1,…,L.

The relationship between the node and its incoming links is represented by a matrix $\mathbf{B} \in \{0,1\}^{N\times L}$, where $B_{il} = 1$ if link l is an incoming link into node i, and $B_{il} = 0$ otherwise. The relationship between the outgoing link and the incoming link of a node is represented by a matrix $\mathbf{C} \in \{0,1\}^{L\times L}$, where $C_{lm} = 1$ if link m is an incoming link into the upstream node of link l, and $C_{lm} = 0$ otherwise. The relationship between the node and its buffer-forwarding outgoing link is represented by a matrix $\mathbf{E} \in \{0,1\}^{N\times L}$, where $E_{il} = 1$ if link l is a buffer-forwarding outgoing link from node i, and $E_{il} = 0$ otherwise. The relationship between the node and its storage-forwarding outgoing link is represented by a matrix $\mathbf{H} \in \{0,1\}^{N\times L}$, where $H_{il} = 1$ if link l is a storage-forwarding outgoing link from node i, and $H_{il} = 0$ otherwise. Suppose there are V videos, denoted as v=1,…,V, being played at the current moment. Each video is allocated with a priority level, denoted as $\alpha_v \left(0 \le \alpha_v \le 1\right)$. A peer can only watch a video at a time. We use a vector $\boldsymbol{\beta} \in \mathbf{R}_+^N$ to denote the peer priority, where $\beta_i = \alpha_v$ if peer i is watching video v, and $\beta_i = 0$ for all other cases. Let p_l denote the packet-loss-ratio (PLR) at link l. We define a link-quality matrix $\mathbf{Q} \in \mathbf{R}_+^{L\times L}$, where $Q_{ll} = 1 - p_l$, and the other components are zero. The peer upload bandwidth is represented by a vector $\mathbf{U} \in \mathbf{R}_+^N$, where U_i is the upload bandwidth of peer i. The peer download bandwidth is represented by a vector $\mathbf{D} \in \mathbf{R}_+^N$, where D_i is the download bandwidth of peer i.

The objective of the optimization problem is to maximize the aggregate weighted throughput taking into account the priority levels of the videos. The objective function is formulated as $f = \boldsymbol{\beta}^T \mathbf{BQX}$, where $\mathbf{X} \in \mathbf{R}_+^L$ is the link rate vector.

We assume that a scalable source coding is used in the P2P streaming, and the maximal source rate is denoted by s_r. If s_r is small, the upload bandwidth in the system will be underutilized. Therefore, we encode the video into a high s_r, such that any peer can receive a rate less than s_r. This is regarded as *source rate constraint*, which is stated mathematically as $\mathbf{BX} \le \mathbf{S}$, where $\mathbf{S} \in \mathbf{R}_{++}^N$ is the source-rate vector with all components equal to s_r. In P2P streaming, bandwidth bottleneck usually occurs at the access links. Therefore, each peer has the *download bandwidth constraint*, formulated as $\mathbf{BX} \le \mathbf{D}$. The buffer-forwarding aggregate outgoing rate can be expressed by $\mathbf{EX}$, and the storage-forwarding aggregate outgoing rate can be expressed by $\mathbf{HX}$. The total outgoing rate, consisting of buffer-forwarding outgoing rate and storage-forwarding outgoing rate, should satisfy the *upload bandwidth constraint*, which is stated as $\mathbf{EX} + \mathbf{HX} \le \mathbf{U}$.

The direct outgoing link from the server should carry a rate not larger than s_r, and each buffer-forwarding outgoing link from peer i (i≠1, node 1 is the server) should carry a rate not larger than the total incoming rate into that peer. This constraint is referred to as *buffer-forwarding constraint*, which can be formulated as $\left(\mathbf{I} - \mathbf{C}\right)\mathbf{X} \le \boldsymbol{\sigma}$, where $\mathbf{I}$ is a $L \times L$ identity matrix, $\boldsymbol{\sigma} \in \mathbf{R}_+^L$ is a buffer-forwarding compensation vector, defined as $\sigma_l = s_r$ if link l is a direct outgoing link from the server or a storage-forwarding link, and $\sigma_l = 0$ otherwise.

We adopt a random replication strategy in which each peer is required to store one random segment. Our algorithm can be easily extended to deal with the case in which multiple segments are stored in each peer. The rate of the stored segment at each peer is represented by a segment-rate vector $\mathbf{F} \in \mathbf{R}_+^N$, where F_i is the rate of the stored segment at peer i. In hybrid-forwarding P2P VoD systems, we have a *storage-forwarding constraint*: the rate of each storage-forwarding outgoing link from each peer should be not larger than the rate of the stored segment. This constraint can be formulated as $\mathbf{X} \le \mathbf{H}^T\mathbf{F} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \in \mathbf{R}_+^L$ is a storage-forwarding compensation vector, where $\xi_l = s_r$ if link l is a buffer-forwarding link, and $\xi_l = 0$ otherwise.

The optimization problem is to maximize the aggregate weighted throughput by optimally determining the link rates, subject to the above five constraints: *source rate constraint*, *download bandwidth constraint*, *upload bandwidth constraint*, *buffer-forwarding constraint*, and *storage-forwarding constraint*. Mathematically, the optimization problem can be formulated as follows.

$$\begin{aligned} \text{maximize } & \boldsymbol{\beta}^T\mathbf{BQX} \\ \text{subject to } & \mathbf{BX} \le \mathbf{S}, \\ & \mathbf{BX} \le \mathbf{D}, \\ & \mathbf{EX} + \mathbf{HX} \le \mathbf{U}, \\ & (\mathbf{I} - \mathbf{C})\mathbf{X} \le \boldsymbol{\sigma}, \\ & \mathbf{X} \le \mathbf{H}^T\mathbf{F} + \boldsymbol{\xi}, \\ & \mathbf{X} \ge \mathbf{0}. \end{aligned} \tag{1}$$

Let $\mathbf{Y} = \min\{\mathbf{S}, \mathbf{D}\}$, $\mathbf{T} = \left[\mathbf{B}^T \ (\mathbf{E}+\mathbf{H})^T \ (\mathbf{I}-\mathbf{C})^T \ \mathbf{I}^T\right]^T$, and $\mathbf{W} = \left[\mathbf{Y}^T \ \mathbf{U}^T \ \boldsymbol{\sigma}^T \ (\mathbf{H}^T\mathbf{F}+\boldsymbol{\xi})^T\right]^T$. We can combine the five constraints into one inequality constraint, and the optimization problem (1) can be rewritten as follows.

$$\begin{aligned} \text{maximize } & \boldsymbol{\beta}^T\mathbf{BQX} \\ \text{subject to } & \mathbf{TX} \le \mathbf{W}, \\ & \mathbf{X} \ge \mathbf{0}. \end{aligned} \tag{2}$$

The optimization problem (2) is linear programming (LP), which can be solved using a centralized algorithm such as the interior point method [6]. In order to develop a distributed algorithm, we change the objective function to a quadratic function. The optimization problem (2) is then changed to:

$$\begin{aligned} & \text{minimize } -\boldsymbol{\beta}^T\mathbf{BQX} + \varepsilon\mathbf{X}^T\mathbf{X} \\ & \text{subject to the constraints in (2),} \end{aligned} \tag{3}$$

where $\varepsilon\,(\varepsilon > 0)$ is a scalar regulation factor. When ε is sufficiently small, the solution to problem (3) is arbitrarily close to the solution to problem (2).

3.2. Distributed solution

The problem in (3) can be solved with a distributed algorithm using dual decomposition. We introduce dual variables $\mathbf{V} \in \mathbf{R}_+^{2N+2L}$ to formulate the Lagrangian corresponding to primal problem (23) as below

$$L(\mathbf{X},\mathbf{V}) = -\boldsymbol{\beta}^T\mathbf{BQX} + \varepsilon\mathbf{X}^T\mathbf{X} + \mathbf{V}^T(\mathbf{TX} - \mathbf{W}). \tag{4}$$

The dual function is the minimum of the Lagrangian [7]:

$$\begin{aligned} G(\mathbf{V}) &= \inf_{\mathbf{X}\ge 0}\{L(\mathbf{X},\mathbf{V})\} \\ &= \inf_{\mathbf{X}\ge 0}\{\varepsilon\mathbf{X}^T\mathbf{X} + (\mathbf{V}^T\mathbf{T} - \boldsymbol{\beta}^T\mathbf{BQ})\mathbf{X}\} - \mathbf{V}^T\mathbf{W}. \end{aligned} \tag{5}$$

The Lagrange dual problem is to maximize the Lagrange dual function. That is: maximize $G(\mathbf{V})$, subject to $\mathbf{V} \ge 0$.

Let $\mathbf{Z} = \mathbf{V}^T\mathbf{T} - \boldsymbol{\beta}^T\mathbf{BQ}$, At the k-th iteration, the link rate vector can be calculated by $\mathbf{X}^{(k)} = \left[-\mathbf{Z}^{(k)}/2\varepsilon\right]^+$, where $[\,\cdot\,]^+$ denotes the projection onto nonnegative real number $\mathbf{R}_+$. Dual variables are updated using subgradient method [8]. The dual variables at the (k+1)-th iteration is given by

$$\mathbf{V}^{(k+1)} = \left[\mathbf{V}^{(k)} - \theta^{(k)}(\mathbf{W} - \mathbf{TX})\right]^+, \tag{6}$$

where $\theta^{(k)}$>0 is the scalar step-size at the k-th iteration.

The proposed algorithm for throughput maximization in hybrid-forwarding P2P VoD systems is fully distributed. Each peer collects the local information through message exchanges with its neighbors to compute the optimal link rates of the incoming links. If there is a topology change due to peer dynamics, the overlay construction module reconstructs the overlay and informs the throughput maximization module, which then updates the dual variables. The change of the dual variables causes the update of the link rates. In this way, both the dual variables and the link rates progressively evolve towards the optimal solution corresponding to the new topology.

4. SIMULATIONS

In the simulation, the maximum source rate is set to 1.3 Mbps. The server has a 2 Mbps upload capacity. There are two classes of peers: cable/DSL peers and Ethernet peers. Cable/DSL peers take 85% of the total population with download capacity uniformly distributed between 0.6 Mbps and 1.0Mbps and upload capacity uniformly distributed between 0.2 Mbps and 0.4 Mbps. Ethernet peers take the remaining 15% of the total population with both upload and download capacities uniformly distributed between 1Mbps and 2 Mbps. The playback time of each peer is uniformly distributed between 0 and the length of the video (1.5 hours). The packet loss rate at each link is uniformly distributed between 0.05 and 0.10. Two videos, denoted as video 1 and video 2, are played at the moment. 60% of the peers are watching video 1, 30% of the peers are watching video 2, and the left peers are idle. Each peer randomly stores one segment. The length of a video segment is fixed at 5 minutes.

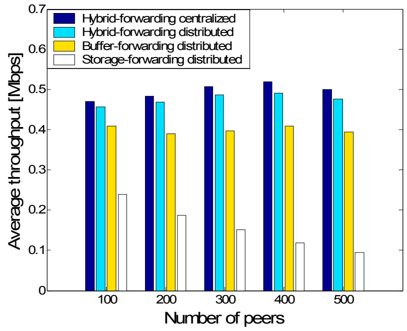

Fig. 2. Comparison of average throughput with different network sizes

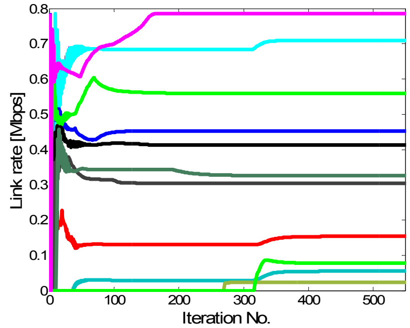

Fig. 3. The iterations of randomly selected 11 link rates in a 200-peer scenario

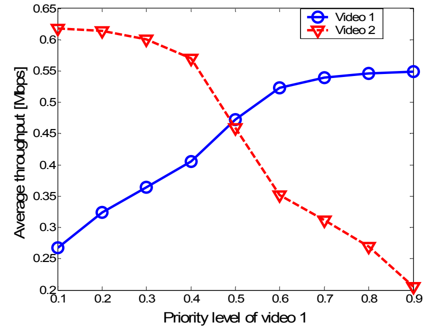

Fig. 4. Average throughput for different priority level in a 200-peer scenario

Fig. 2 compares the average throughput among different schemes. "Hybrid-forwarding centralized" represents the results obtained by solving the optimization problem (2) with a centralized interior point method; "Hybrid-forwarding distributed" represents the results obtained with the proposed distributed algorithm; "Buffer-forwarding distributed" represents the solution obtained by a distributed algorithm in the P2P VoD system with only buffer-forwarding links; "Storage-forwarding distributed" represents the solution obtained by a distributed algorithm in the P2P VoD system with only storage-forwarding links. "Hybrid-forwarding centralized" obtains globally optimal solution in hybrid-forwarding P2P VoD. However this scheme requires global information, thus not suitable for P2P applications. The average throughput obtained with the proposed algorithm, "Hybrid-forwarding distributed", is very close to the globally optimal throughput. By sharing both buffer and storage among the P2P community, the proposed hybrid-forwarding scheme improves the throughput by 12% – 23% over the scheme with only buffer-forwarding links and by at least 90% over the scheme with only storage-forwarding links.

We jointly optimize 659 link rate variables in a 200-peer P2P VoD scenario. All the 659 link rates start from an initial value 0, and converge to their optimal values within 549 iterations. For clear presentation, we just randomly choose 11 link rate variables and show their iterations in Fig. 3.

In hybrid-forwarding P2P VoD systems, we can implement the service differentiation among videos by adjusting the priority level of each video. We vary the priority level α_1 of video 1 from 0.1 to 0.9, and the priority level α_2 of the video 2 is set by $\alpha_2 = 1 - \alpha_1$. As shown in Fig. 4, the average throughput of video-1 viewers is increased as the priority level of video 1 goes up. When the video-1 priority level is higher than video-2 priority level, video-1 viewers are allocated with a larger portion of the upload bandwidth, thus achieving a larger throughput compared to the video-2 viewers.

4. CONCLUSIONS

In this paper, we propose a hybrid-forwarding P2P VoD architecture, which integrates both the buffer-forwarding approach and storage-forwarding approach. We also develop a distributed algorithm to maximize the throughput. Simulation results demonstrate that the proposed hybrid-forwarding architecture achieves a much higher throughput compared to the architecture with only buffer-forwarding links or with only storage-forwarding links.

5. REFERENCES

[1] X. Zhang, J. Liu, B. Li, and T. P. Yum, "CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming," in *Proc. of IEEE INFOCOM*, vol. 3, pp. 2102–2111, Mar. 2005.

[2] Z. Li and A. Mahanti, "A Progressive Flow Auction Approach for Low-Cost On-Demand P2P Media Streaming," in *Proc. of QShine*, Aug. 2006.

[3] W. P. Yiu, X. Jin and S. H. Chan, "Distributed storage to support user interactivity in peer-to-peer video streaming," in *Proc. of IEEE ICC*, vol. 1, pp. 55–60, Jun. 2006.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204-1216, Jul. 2000.

[5] Y. Cui, B. Li, and K. Nahrstedt, "oStream: asynchronous streaming multicast," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 91–106, Jan. 2004.

[6] R. J. Vanderbei, *Linear programming: foundations and extensions*, 2nd Edition, Springer Press, 2001.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[8] D. Palomar and M. Chiang, "A tutorial on decomposition methods and distributed network resource allocation," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[9] X. Xu, Y. Wang, S. P. Panwar, and K. W. Ross, "A Peer-to-Peer Video-on-Demand System using Multiple Description Coding and Server Diversity," in *Proc. of IEEE ICIP*, vol. 3, pp. 1759–1762, Oct. 2004.