<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Efficient and flexible inter-overlay scheduling of media streams for multi-channel P2P streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Jin, X; Kwok, YK</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>The 2012 International Conference on Computing, Networking, and Communications (ICNC 2012), Maui, HI., 30 January-2 February 2012. In Proceedings of ICNC, 2012, p. 820-824</td>
</tr>
<tr>
<td><strong>Issued Date</strong></td>
<td>2012</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10722/160276">http://hdl.handle.net/10722/160276</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>International Conference on Computing, Networking and Communications Proceedings. Copyright © IEEE.; ©2012 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.; This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.</td>
</tr>
</tbody>
</table>
Efficient and Flexible Inter-Overlay Scheduling of Media Streams for Multi-Channel P2P Streaming

Xin Jin and Yu-Kwong Kwok
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam, Hong Kong SAR

Abstract—Existing studies on channel bandwidth imbalance in P2P multi-channel streaming systems have been exclusively focused on inter-overlay bandwidth allocation. However, an efficient inter-overlay scheduling algorithm is still in lack for benefactors. To this end, this paper presents an inter-overlay substring scheduling algorithm compatible with various overlay meshes for active inter-overlay cooperation, through which the outbound bandwidth of benefactors can be efficiently utilized and bandwidth-deprived channels receiving benefactions can attain a better streaming quality.

Index Terms—Peer-to-peer multimedia streaming, active multichannel cooperation, inter-overlay chunk scheduling, unstructured meshes

I. INTRODUCTION

The strength of peer-to-peer (P2P) design, compared with the client-server architecture, is the capacity to efficiently explore distributed client resources and henceforth reduce the economic cost incurred to media service providers. Specifically, peers participating in the same channel formulate a swarm and reciprocate each other by scheduling chunks cooperatively. Hereafter, the terms overlay and channel are used interchangeably to represent the dissemination of one specific media content.

However, channel heterogeneity, incurred by autonomous peer participation, inevitably leads to disparate service quality experienced by various channels. The situation is further aggravated in relatively unpopular channels because statistically enough bandwidth provision cannot be consistently guaranteed [1]. In particular, measurement study of PPLive reveals significant bandwidth supply variations in diverse channels [2].

Thus, an intuitive idea is that peers within a bandwidth-rich channel \( c_r \in C \) (\( C \) is the set of channels existing in the system) altruistically join a bandwidth-deprived channel \( c_d \in C \) and serve as benefactors (i.e., benefit recipients) by feeding residential peers of \( c_d \) (\( C_d \) is the secondary overlay of benefactors). To make this happen, the following three open questions should be properly answered for a peer \( p \) in \( c_r \):

- **Whether to serve as a benefactor and if so, which channel to benefit?** Such choices depend on the bandwidth availability of peer \( p \). Moreover, considering peer rationality, incentive provision for benefactors is desired to motivate inter-overlay contribution.
- **How to allocate its outbound bandwidth?** Benefactor \( p \) has to resolve the bandwidth allocation between \( c_r \) and \( c_d \) — an essential step to optimally reallocate available outbound bandwidth resources existing in various channels. This dictates the streaming quality of both \( c_r \) and \( c_d \).
- **How to efficiently schedule chunks?** That is, how to schedule chunks so that the outbound bandwidth allocated to \( c_d \) is fully exploited to maximally enhance the streaming quality of channel \( c_r \) in this paper, we term the chunk scheduling operation of benefactors to help and relay chunks to residential peers inter-overlay scheduling. Correspondingly, the chunk scheduling operation of residential peers is referred to as intra-overlay scheduling.

Several studies [2], [3] exist in the literature to study the bandwidth allocation problem. However, there is no such an inter-overlay scheduling algorithm with the flexibility to couple with various intra-overlay scheduling designs. To this end, this paper is to design a robust and configurable inter-overlay scheduling algorithm with both such flexibility and the bandwidth utilization efficiency to fully leverage allocated outbound bandwidth for the purpose of inter-overlay benefaction. To the best of our knowledge, this is the first attempt to achieve this design objective. The non-trivial difference between inter-overlay scheduling and intra-overlay scheduling implies the design difficulty.

The reminder of this paper proceeds as follows. Section II presents recent advances in P2P multimedia streaming. Section III formulates the inter-overlay scheduling problem and illustrates our proposed inter-overlay substring scheduling protocol. In Section IV, we present performance evaluation methodology. Simulation results are presented in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

There exists a large body of literature on mesh based P2P overlays and intra-overlay scheduling has been extensively investigated [4], [5]. Specifically, peer \( i \) determines how to select neighbor \( j \) to request chunk \( a \) from the set of neighbors possessing chunk \( a \notin C_i \), namely \( N_i^a = \{p|(p \in N_i) \land (a \in C_p)\} \), where \( N_i \) is the neighbor set of peer \( i \) and \( C_p \) is the set of chunks in \( p \)'s buffer.

- **Random.** Peer \( i \) randomly selects \( j \in N_i^a \).
- **Least Load** [4]. Peer \( i \) selects \( j \) with the maximum value of outbound bandwidth over sending queue size. That is,

\[
    j = \arg \max_{p \in N_i^a} \frac{O_p}{Q_p},
\]
where $l_j$ is the sending queue length of peer $j$.

- **Bandwidth Aware** [5]. The probability that $i$ selects $j$ is
  \[
  \pi_i(j) = \frac{O_j}{U_i}, \quad U_i = \sum_{p \in N_i^p} O_p.
  \]

However, not until recently, inter-overlay cooperation has become an important topic to improve system performance. Liao et al. [6] propose to utilize inter-overlay path optimization by constructing efficient paths using peers from different content overlays. An auction based mechanism [3] is designed by Wu et al. to resolve conflicts among coexisting streaming overlays for efficient bandwidth allocation. Wu et al. [2] implement the idea of View-Upload-Decoupling (VUD) by dividing media content into independent substreams, whose design resembles multiple tree streaming. Peers in VUD are required to retrieve a substream from a non-residential channel for uploading purposes. A theoretical study based on queueing theory is performed in [1], further demonstrating the strength of inter-overlay design. Recently, Wang et al. [7] formulate inter-overlay bandwidth sharing as a utility maximization problem, performed distributively by each peer.

III. INTER-OVERLAY SCHEDULING: DESIGNS AND IMPLEMENTATIONS

A. Problem Formulation

The impact of bandwidth provision. Bandwidth provision is critical for channel streaming quality. Thus, we firstly define resource index $\rho$ to quantitatively measure the outbound bandwidth provision of channel $c$:
\[
\rho_c = \frac{\sum_{p \in \mathcal{P}_c} O_p + O_c^s}{|\mathcal{P}_c| \cdot s_c},
\]
where $\mathcal{P}_c$ is the set of peers participating in channel $c \in \mathcal{C}$, $O_p$ and $O_c^s$ are respectively the outbound bandwidth of peer $p \in \mathcal{P}_c$ and the server of channel $c$. Denote by $\mathcal{C}$ and $s_c$ respectively the set of coexisting channels and the streaming rate of channel $c$. Similarly, the resource index after inter-overlay benefitation is
\[
\rho^b_c = \frac{\sum_{b \in \mathcal{B}_c} O_b + \sum_{p \in \mathcal{P}_c} O_p + O_c^s}{|\mathcal{P}_c| \cdot s_c},
\]
where $\mathcal{B}_c$ is the set of benefactors residing in channel $c$. Note that we have not counted in the outbound bandwidth overhead incurred by benefactors. We omit the subscript $c$ in the following discussion. Then, the bandwidth benefitation obtained from benefactors can be quantitatively measured by
\[
h = \rho^b - \rho.
\]

To illustrate the impact of channel bandwidth provision on streaming quality, we implement the least load scheduling algorithm with the capacity to achieve near-optimal streaming quality in systems with enough bandwidth provision [4]. Streaming quality in terms of delivery ratio under different resource index scenarios is shown in Fig. 1(a) and Fig. 1(b) with the simulation environment depicted in Section IV. Obviously, better streaming quality can be attained with the increase of resource index and bandwidth deficit significantly degrades the streaming quality. Consequently, we can learn the importance of inter-overlay bandwidth sharing and the necessity of inter-overlay scheduling.

Design objective of inter-overlay scheduling. However, inter-overlay scheduling is different from intra-overlay scheduling in that benefaction peers do not individually need chunks retrieved from the secondary overlay. This inevitably results in *outbound bandwidth overhead* incurred by chunk retrieval before the possibility of bandwidth benefitation. Notice that it is not the signalling overhead for overlay maintenance, but the bandwidth consumption of peers to upload chunks to benefactors in the secondary overlay.

Thus, the optimal inter-overlay scheduling problem can be formally modeled as
\[
\max \{ \sum_{b \in \mathcal{B}} (u_b - d_b) \},
\]
subject to
\[
d_b \leq D_b, \quad u_b \leq O_b,
\]
where $D_b$ and $O_b$ are respectively the inbound bandwidth limit and the outbound bandwidth limit allocated to benefactor $b$ for inter-overlay benefitations. $d_b$ and $u_b$ respectively represent the actual inbound and outbound consumption, and $\mathcal{B}$ is the total set of benefactors existing in the system. Simply put, it is desirable for benefactors to download less while contributing more in the SO. If each benefaction peer possesses perfect information about the demand and supply of each chunk, peers can solve the above problem by recursively retrieving chunks with the highest demand. However, network decentralization renders this problem extremely difficult.

In this paper, to evaluate the bandwidth utilization efficiency of inter-overlay scheduling, we define the *benefaction ratio* for each benefactor $b$
\[
R_b = \frac{u_b}{d_b}, \quad \forall b \in \mathcal{B}.
\]
Considering peer dynamics and the fact that $R_b$ depends on both chunk availability and chunk demands in the neighborhood of benefactors, we resort to heuristics for inter-overlay scheduling design.

B. Inter-Overlay Substream Scheduling

An intuitive idea is to leverage on substream scheduling, because peers only need to retrieve part of the stream for
benefaction purposes. Due to both our assumption that outgoing bandwidth constraint is the main bottleneck faced by resource-poor overlays and the fact that pull-scheduling can complement the push mode by retrieving chunks not timely pushed and further enhancing the streaming quality [8], we utilize push-pull inter-overlay substream scheduling to achieve a high benefaction ratio.

To minimize the outbound bandwidth overhead, each benefactor subscribes to one single overlay and relays the received chunks to its neighbors. Each benefactor randomly selects a substream when joining the bandwidth-deprived channel and retrieves the corresponding chunks in the very substream. The download sliding windows of benefactors are updated in buffer map exchanges just as residential peers. As discussed above, we utilize the push-pull mode, which not only integrates the push-mode compatible with the assumption that the uplink-constraint is the bottleneck, but also adopts the pull-mode as a fallback strategy.

Substream $k$ is defined as the set of chunks with sequence number $S_n$ satisfying

$$S_n \% K = k,$$

where $K$ is the total number of substreams existing in the system. Upon receiving a chunk from a benefactor, the residential peer will subscribe to the benefactor. Each benefactor will push chunks to all the neighbors subscribing to its substream. And the neighbor degree of benefactor $b$ is determined by

$$Nbr = \alpha \cdot K \cdot \frac{O_b}{s},$$

where $Nbr$ is the neighbor degree, $K$ is the number of substreams, and $s$ is the streaming rate. System parameter $\alpha$ can be utilized to adjust the neighbor degree.

In our design, residential peers randomly select its neighbors from both residential peers and benefactors. However, benefactors’ neighborhood only consists of residential peers in that their task is to help residential peers instead of benefactors themselves. Moreover, only if two benefactors share the same substream number, can they reciprocate to each other.

Compared with VUD [2], our design can achieve lower outbound bandwidth overhead due to the fact that VUD requires every peer retrieve one substream from non-residential channels, while our design only desires such action from benefactors. Moreover, our design possesses the flexibility to fit into various chunk-based streaming systems.

IV. PERFORMANCE EVALUATION

We now proceed to evaluate the inter-overlay substream scheduling algorithm. In this section, we respectively describe the simulation setting, simulation methodology and metrics utilized to evaluate system performance.

A. Simulation Setting

We explore and evaluate the inter-overlay substream scheduling algorithm stated above through simulations, based on the P2P media streaming simulator originally developed by Zhang [9] for intra-overlay scheduling. This is an event-driven packet-level C++-based simulator with the capacity to simulate a maximum of 10,000 peers simultaneously joining a single overlay.

Unless otherwise stated, simulation results correspond to the simulation setting given in Table I. Note that the setup of the three types of nodes is for residential peers. The resource index $\rho$ without bandwidth benefactor is 0.9.

<table>
<thead>
<tr>
<th>TABLE I SIMULATION SETTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor Size of Residential Peers</td>
</tr>
<tr>
<td>Neighbor Size of Benefactors</td>
</tr>
<tr>
<td>Streaming Rate</td>
</tr>
<tr>
<td>Server Upload Capacity</td>
</tr>
<tr>
<td>Type 1 Node Capacity (Up/Down)</td>
</tr>
<tr>
<td>Type 2 Node Capacity (Up/Down)</td>
</tr>
<tr>
<td>Type 3 Node Capacity (Up/Down)</td>
</tr>
<tr>
<td>Type 1 Node Fraction</td>
</tr>
<tr>
<td>Type 2 Node Fraction</td>
</tr>
<tr>
<td>Type 3 Node Fraction</td>
</tr>
<tr>
<td>Residential Peer Population</td>
</tr>
<tr>
<td>Benefactor Population</td>
</tr>
<tr>
<td>Intra-Overlap Scheduling</td>
</tr>
</tbody>
</table>

B. Simulation Methodology

We modify the simulator by deploying two kinds of peers in the simulator, one group of which represents the residential peers and the other group joins in the system as benefactors. Residential peers follow intra-overlay scheduling method to retrieve chunks they need. On the other hand, benefactors take the proposed inter-overlay scheduling strategy. Initially, we set the number of substreams to be 10 for the study of the algorithm’s effectiveness to improve streaming quality and the impact of graph degree, peer dynamics, different intra-overlay scheduling schemes, and benefactor heterogeneity on system performance. Then, we vary the number of substreams to study its influence on system design.

Static and dynamic simulation environments. In a static environment, nodes, including both residential peers and benefactors, join in one by one in the initialization period. After that, they will persist in the life time. Dynamics refer to peer churn (i.e., peer joining/leaving/failure, etc). In the dynamic environment, each node repeatedly joins and departs the system. The online and offline durations of each node per time are exponentially distributed, which incurs a high peer churn rate with an average online duration of 20 minutes.

Benefactor heterogeneity. We utilize the 3-class scenario to evaluate whether the inter-overlay scheduling logic still performs well when the overlay is built upon a heterogeneous network. The 3-class scenario is based on a 3-class bandwidth distribution: low-bandwidth peers with outbound bandwidth 0.5$\overline{B}$, mid-bandwidth peers with outbound bandwidth $\overline{B}$, and high-bandwidth peers with outbound bandwidth $2\overline{B}$. The fraction of these 3 classes of peers is respectively $t/3$, $1-t$, and $2t/3$ from low to high, where $t$ is the heterogeneity factor and $0 \leq t \leq 1$; thus, the average bandwidth is $2\overline{B} \cdot t/3 + \overline{B} \cdot (1-t) + 0.5\overline{B} \cdot 2t/3$. This scenario captures the most important properties of heterogeneous networks and has already been widely utilized in the literature [10].
Neighborhood. We also study the impact of the neighborhood degree of beneficial peers on streaming quality. To focus on the impact of graph degree, the beneficial peers are homogeneous with outbound bandwidth 270 Kbps and we hold the intra-overlay scheduling algorithm to be push-pull. We also hold the neighbor degree of residential peers constant (i.e., 20).

C. Performance Metrics

We define the following metrics to measure the streaming quality of peers:

- **Distribution delay.** By distribution delay, we mean the elapsed time from the instant a chunk is generated by the source to the moment it is received by a peer.
- **Delivery ratio.** Aside from distribution delay, to quantitatively measure the streaming quality, we define delivery ratio to represent the number of chunks arriving at each node before or on playback deadline over the total number of chunks that each node should receive.

V. SIMULATION RESULTS

In this section, we present our simulation results following the simulation methodology described above.

A. The Effectiveness of Inter-Overlay Substream Scheduling

We study the effectiveness of bandwidth benefit via the delivery ratio distributions under different values of $h$ (cf. Section III). As a comparison baseline, we also provide the delivery ratio distribution for intra-overlay scheduling with $\rho = 1.1$. Fig. 2(a) shows that after inter-overlay bandwidth benefit, system performance can be improved.

Fig. 2(b) shows that our substream inter-overlay scheduling scheme can also improve the system performance with pull intra-overlay scheduling. However, to achieve similar streaming performance, the pull scheme requires more bandwidth benefit than the push-pull scheme. Indeed, the push-pull intra-overlay scheduling scheme can leverage the system bandwidth more efficiently. In Fig 3(a), we explicitly compare the average delivery ratio when we utilize the two intra-overlay schemes under different values of $h$. It reveals that our inter-overlay scheduling couples better with push-pull intra-overlay scheduling.

B. The Impact of Intra-Overlay Peer Selection Methods.

Fig. 3(b) compares three different peer selection schemes used for intra-overlay scheduling via distribution delay: random, least load, and bandwidth aware. The simulation setup is $\alpha = 3.0$ and $h = 0.3$. The average distribution delay of these three schemes are respectively $1649ms$ (random), $1522ms$ (capacity aware), and $1262ms$ (bandwidth aware). The bandwidth aware scheme performs best in terms of distribution delay in that the node heterogeneity is properly considered.

C. The Impact of Graph Degree.

Fig 4(a) illustrates that initially the streaming quality improves in terms of both delivery ratio and distribution delay with the increase of the neighbor degree of beneficial peers. However, when the neighbor degree is large enough, the streaming quality degrades with the increase of neighbor degree. This may result from the fact that when neighbor degree is small, beneficial peers cannot effectively relay chunks to enough residential peers. Then, the increase in neighbor degree can improve the streaming quality. However, when the neighbor degree is large enough, the proportion of benefactor neighbors of residential peers increases, which may affect effective intra-overlay scheduling, considering the fixed neighbor degree of residential peers. Fig 4(b) illustrates the variation of benefit ratio with respect to $\alpha$, which shows that the benefit ratio of the system is around 3.8.

D. The Impact of Dynamic Environment.

To further study the performance of the inter-overlay scheduling substream algorithm, we compare the scheduling performance under both static and dynamic environment. We setup the following parameters: $\alpha = 3.0$, and $h = 0.3$. Fig. 5(a) shows us the comparison of delivery ratio and Fig. 5(b) shows us the comparison of distribution delay. The average distribution delays in static and dynamic environments are respectively $1649ms$ and $4245ms$. We observe that the inter-overlay substream scheduling algorithm still performs well in the dynamic environment.
the worst performance when the distribution of residential peers has not been reconfigured in the study on the effects of benefactor heterogeneity, the bandwidth allocation schemes are in prospect.

VI. CONCLUSION

In this paper, we present an inter-overlay scheduling substream algorithm for active benefactors in multichannel P2P multimedia systems. Extensive simulations demonstrates its efficacy to improve the streaming quality of bandwidth-deprived channels. We also study the impact of graph degree, peer dynamics, benefactor heterogeneity and the substream number on system performance.

Our proposed algorithm can be flexibly coupled with existing P2P overlay meshes, as verified via simulation of different intra-overlay scheduling algorithms. This proves its efficacy to assist existing inter-overlay bandwidth sharing schemes to further enhance bandwidth utilization efficiency. Moreover, we propose benefaction ratio to evaluate the bandwidth utilization efficiency. Incentive provision for benefactors and more efficient bandwidth allocation schemes are in prospect.

REFERENCES