# On Game Theoretic Peer Selection for Resilient Peer-to-Peer Media Streaming

Mark Kai Ho Yeung, *Student Member*, *IEEE*, and Yu-Kwong Kwok, *Senior Member*, *IEEE*

**Abstract**—Peer-to-peer (P2P) media streaming quickly emerges as an important application over the Internet. A plethora of approaches have been suggested and implemented to support P2P media streaming. In our study, we first classified existing approaches and studied their characteristics by looking at three important quantities: number of upstream peers (parents), number of downstream peers (children), and average number of links per peer. In existing approaches, peers are assigned with a fixed number of parents without regard to their contributions, measured by the amount of outgoing bandwidths. Obviously, this is an undesirable arrangement as it leads to highly inefficient use of the P2P links. This observation motivates us to model the peer selection process as a cooperative game among peers. This results in a novel peer selection protocol such that the number of upstream peers of a peer is related to its outgoing bandwidth. Specifically, peers with larger outgoing bandwidth are given more parents, which make them less vulnerable to peer dynamics. Simulation results show that the proposed protocol improves delivery ratio using similar number of links per peer, comparing with existing approaches under a wide range of system parameters.

**Index Terms**—P2P media streaming, coalition, incentives, cooperative game, structured P2P topology, unstructured P2P topology, peer selection.

## 1 Introduction

As the Internet has become a core part of our daily lives, people rely more and more on it for information retrieval and exchange. One recent profound trend is media distribution over the Internet. Of particular importance is peer-to-peer (P2P) streaming of media data due to the open architecture of the Internet [1], [2], [3], [14], [22]. In a typical P2P media streaming scenario, a media server could support tens of thousands of concurrent users [19], [30], [31]. Such a high scalability is achieved by having the server stream media data to a subset of the whole client population, which in turn shares the media data with other clients in a P2P manner [11], [12], [13], [15], [16], [19], [20], [21], [25], [30], [31].

Existing P2P media streaming approaches can be broadly classified into two categories: structured and unstructured. For the structured category, the direction of flow of packets containing media information is fixed after a particular peer joins the media distribution overlay network. On the other hand, peers can exchange media packets with their neighbors in both directions for the unstructured category [27]. We can further divide the unstructured category into pure and hybrid approaches. In this study, we report our results on the structured case. The reader is referred to [27] for our results on the unstructured case. In structured approaches, peers can be organized in three different logical structures: a single tree, multiple trees, and a directed acyclic graph (DAG). We can analyze the characteristics of existing approaches by looking at three important quantities, namely, number of upstream peers, number of downstream peers, and average number of links per peers.

Unfortunately, existing P2P media streaming approaches do not fully exploit peer heterogeneity in the design of the P2P network formation protocols. Specifically, the number of upstream peers (parents) assigned to a particular peer is usually fixed without regard to its outgoing bandwidth. Obviously, this rigid arrangement is undesirable as it can potentially lead to inefficient P2P links.

Our major contribution in this paper is that we model the peer selection process as a cooperative game, called the *peer selection game*, in which peers are free to form coalitions for media streaming. We devise a set of specific conditions for stable coalitions. Based on our novel game theoretic analysis, we propose a novel peer selection protocol, where each peer judiciously selects an appropriate number of parents depending on its outgoing bandwidth. In particular, the proposed protocol is robust in that it autonomously assigns more parents to peers with larger outgoing bandwidth, making them less susceptible to peer dynamics, i.e., joining and leaving events (formally defined in Section 2). This is because they would accept more downstream peers (children) and, thus, are more important entities. On the other hand, peers having smaller outgoing bandwidth are given fewer children to make the average number of links per peer comparable to existing approaches. This arrangement also provides participating peers with incentives to contribute more resources because increasing the amount of outgoing bandwidth implies a lower likelihood for them to be affected by peer dynamics. Simulation results show that the proposed game theoretic peer selection protocol improves delivery ratio using comparable number of links per peer, comparing with existing approaches in many diverse scenarios.

- *M.K.H. Yeung is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: khyeung@eee.hku.hk.*
- *Y.-K. Kwok is with the Electrical and Computer Engineering, Colorado State University, B112 Engineering Building, 1373 Campus Delivery, Fort Collins, CO 80523-1373. E-mail: Ricky.Kwok@colostate.edu.*

*Manuscript received 5 Oct. 2007; revised 4 Oct. 2008; accepted 2 Dec. 2008; published online 5 Dec. 2008.*
*Recommended for acceptance by K. Hwang.*
*For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2007-10-0354.*
*Digital Object Identifier no. 10.1109/TPDS.2008.253.*

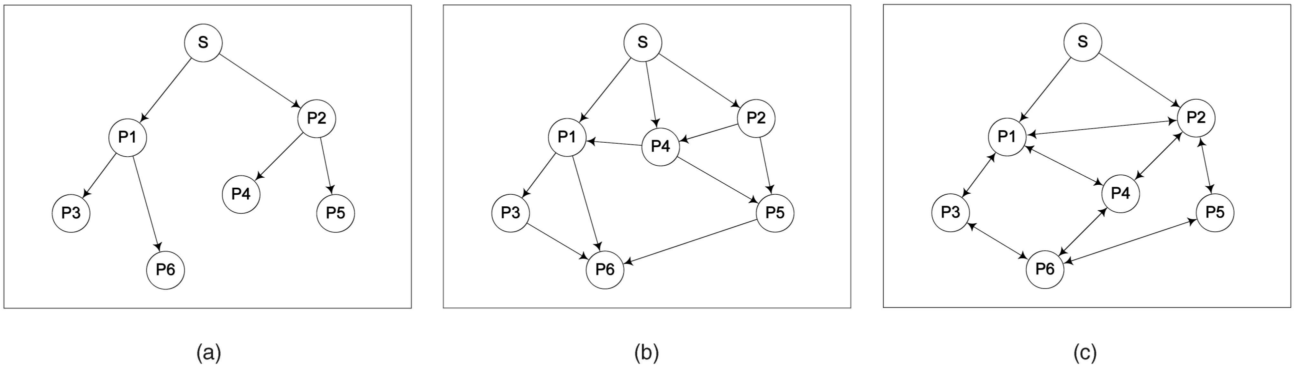


Fig. 1. Logical topology. (a) Tree. (b) DAG. (c) Unstructured.

## 2 Overview of Existing P2P Media Streaming Approaches

We consider a media streaming application with a large audience. It is assumed that the content provider, or the media server, has insufficient resources to serve all the requests simultaneously. The use of P2P technologies to deliver media content is a low-cost and scalable method to satisfy the overwhelming demands. In a P2P media streaming system, there are three major entities: media content, server nodes (server), and peer nodes (peer). Their functions are as follows:

- The *media content* is the target audio and video resources to be disseminated. It is encoded in the constant bit rate (CBR) format at the rate of $r$ Kbps. The server divides the media into a stream of equally sized packets and delivers them to the peers. Each packet contains some media information, e.g., a media frame. We assume that the quality perceived by a peer is determined by the number of received packets.
- A server node, or simply *server*, is the content provider that owns the media content. It can be a dedicated workstation or a content distribution network. However, the aggregate outgoing bandwidth of servers cannot satisfy all streaming requests. Each server is responsible for distributing media packets to a subset of peer nodes. We assume that the core network is not the performance bottleneck.
- A peer node, or simply *peer*, consumes the media content. We assume that the incoming bandwidth of all peers is larger than the media rate, i.e., $\geq r$. On the other hand, each peer can freely determine the amount of outgoing bandwidth to contribute. Specifically, we denote the amount of outgoing bandwidth contributed by peer $x$ as $b_x$.

There have been many different approaches suggested to organize peers for P2P media streaming. We broadly classify them into two main categories: structured and unstructured. Within the structured category, each peer is assigned with dedicated upstream peers (parents) and dedicated downstream peers (children). We can further divide the structured category into three approaches: single tree, multiple trees and DAG. On the other hand, there is no such restriction for the unstructured category. Peers can exchange media packets with their neighbors in both directions. We discuss the characteristics of existing P2P media streaming approaches in the following.

In the single tree approach [7], [8], [11], [19], [20], [21], peers are organized in a logical tree structure rooted at the server, as shown in Fig. 1a. Each peer is assigned with one upstream peer as its parent. Each peer depends on the sole parent for all media packets originated from the server. Peer is also obliged to accept some downstream peers as their children. The number of downstream peers is determined by the peer's outgoing bandwidth. Since each child requires $r$ Kbps from its parent, the number of possible downstream peers accepted by peer $x$ is $\lfloor \frac{b_x}{r} \rfloor$. We denote the single tree approach as Tree(1) and summarize its characteristics mathematically by the following equations:

$$\text{Number of upstream peers in Tree}(1) = 1, \tag{1}$$

$$\text{Number of downstream peers in Tree}(1) = \left\lfloor \frac{b_x}{r} \right\rfloor, \tag{2}$$

$$\text{Number of links per peer in Tree}(1) = O(1). \tag{3}$$

As each peer relies on its sole parent for all media packets, the single tree approach is vulnerable to *peer dynamics*, which is formally defined as the turn-over rate per unit time of the participants. Furthermore, the outgoing bandwidth of peers may not be fully utilized because each increment costs $r$ Kbps.

Similarly, the multiple trees approach [5], [12] is also based on the tree structure. However, peers are organized in more than one tree to be more resilient to peer dynamics. Specifically, the server uses multiple descriptive coding (MDC) [9], [18], which allows media packets to be delivered in $k$ independent streams. These different packet streams are considered as different "descriptions" of the original video. A receiver can always recover the original video with certain distortions as governed by the total number of packets received. This is a salient feature because the recovered video quality is independent of specific packets but rather just depends on the amount of information received. This is in stark contrast with previous approaches such as scalable layered coding.

Now, according to MDC, our model is such that each of the $k$ streams is handled by a separate tree rooted at the server. Peers can obtain the complete media content by joining the $k$ trees. As such, this increases the number of upstream peers to $k$, while the number of possible downstream peers becomes $\lfloor \frac{b_x}{r/k} \rfloor$. We denote the multiple trees approach as $\text{Tree}(k)$ and summarize its characteristics mathematically by the following equations:

$$\text{Number of upstream peers in Tree}(k) = k, \tag{4}$$

$$\text{Number of downstream peers in Tree}(k) = \left\lfloor \frac{b_x}{r/k} \right\rfloor, \tag{5}$$

$$\text{Number of links per peer in Tree}(k) = O(k). \tag{6}$$

The multiple trees approach is less susceptible to peer dynamics because each peer can obtain media packets from $k$ different parents. It also improves the utilization of peer outgoing bandwidth because each increment is reduced to $\frac{r}{k}$Kbps. The major drawback is the increased overheads because peers have to maintain $k$ independent trees. Another potential drawback is that the actual media rate may be slightly increased due to the less efficient coding scheme.

More recently, the DAG approach [13], [16] has been proposed to achieve resilience without the need to maintain more than one structure. Instead of trees, peers are organized in a DAG, as shown in Fig. 1b. This allows peers to control the number of parents and children for media streaming. For example, each peer is assigned with $i$ parents and $j$ children. We denote the DAG approach as $\text{DAG}(i, j)$ and summarize its characteristics mathematically by the following equations:

$$\text{Number of upstream peers in DAG}(i, j) = i, \tag{7}$$

$$\text{Number of downstream peers in DAG}(i, j) = j, \tag{8}$$

$$\text{Number of links per peer in DAG}(i, j) = O(i). \tag{9}$$

As can be seen, the DAG approach is a generalization of the multiple trees approach. Peers are allowed to have $i$ parents, which improves resilience to peer dynamics. On the other hand, the server does not need to split media packets but delivers them in one single stream. This avoids the increased overheads in maintaining multiple structures. The major challenge is to ensure that there is no loop among peers. To avoid loops, it has been proposed that peers when accepting a new peer should make sure that the new peer is not in its upstream. This checking process inevitably increases the overhead in maintaining the P2P structure.

The above structured approaches organize peers differently for media streaming. Each peer receives media packets from some upstream peers and forward media packets to some other downstream peers. In other words, the flow of media packets is fixed when a new peer joins the P2P media streaming network. On the other hand, in the unstructured category [27], peers can exchange media packets with the same set of peers, called neighbors. Specifically, in an unstructured model, peers are organized in a random graph, as shown in Fig. 1c. Here, each peer is assigned with $n$ neighbors, and the availability of media packets determines the direction of media packets. We denote the pure unstructured approach as $\text{Unstruct}(n)$ and summarize its characteristics mathematically by the following equations:

$$\text{Number of upstream peers in Unstruct}(n) = n, \tag{10}$$

$$\text{Number of downstream peers in Unstruct}(n) = n, \tag{11}$$

$$\text{Number of links per peer in Unstruct}(n) = O(n). \tag{12}$$

Each peer is connected to its neighbors and does not require to maintain any specific networkwide structure. This improves the scalability of P2P media streaming. However, the number of neighbors, $n$, should be carefully chosen to ensure the resultant random graph is connected. It has been shown in [26] that $n$ should be at least $0.5139 \log(|N|)$, where $|N|$ represents the total number of peers, for connectedness with high probability. For example, we should set $n = 5$ when there are 5,000 peers. However, the unstructured approach can potentially lead to a larger delay because media packets are randomly traversed among peers. Furthermore, the unstructured approach requires each peer to have a larger buffer to cater for the randomness in peer connectivity. In practice, this is usually not a significant concern because storage size is often not a limiting factor and most P2P media streaming applications focus on streaming stored media content. On the other hand, the hybrid unstructured approach combines the use of a structured approach with the unstructured approach [23], [24].

From the above discussion, we can see that existing approaches restrict the number of upstream peers using various system parameters. The restriction is to limit the average number of links maintained by each peer, which is a measure of communication overheads involved in P2P media streaming. However, we find that such restriction is indeed unnecessary and does not take the peer heterogeneity into consideration. Since peers are allowed to determine the amount of outgoing bandwidth to contribute, it would be more desirable to judiciously allocate upstream and downstream peers with respect to a peer's contribution. For example, peers with larger outgoing bandwidth are expected to have more downstream peers. We should provide these more capable peers with more upstream peers so that they are more resilient to peer dynamics. On the other hand, peers with smaller outgoing bandwidth can be assigned with fewer upstream peers to reduce their overheads. This arrangement serves as an incentive measure for peers to contribute. Based on the structured DAG approach, we model the peer selection process as a cooperative game, which strives to handle heterogeneous peers and achieve the above characteristics.

## 3 Proposed Game Theoretic Peer Selection Approach

In P2P media streaming, each peer can choose its upstream peers (parents) and downstream peers (children). We consider peers as rational entities and model the peer selection process as a strategic game.

We first focus on the case where there is only one parent, $p$, and a set of children, $c_1, c_2, \ldots, c_n$. We would like to study how $p$ should select its children such that the resultant parent-child relationships are stable and resilient to peer dynamics. Specifically, we formulate a cooperative game where the players are the parent and its children. The objective is to form a stable coalition which creates the highest aggregate value. Here, stability is defined as the probability that a participant departs from the coalition and acts alone. The aggregate value is to be distributed among the members. In other words, we have to tackle two inter-related issues: 1) formation of a stable coalition; and 2) distribution of the aggregate value. We define the elements of cooperative game and what constitutes a stable coalition as follows:

A cooperative game consists of a finite set of players, $N$, and a scalar-valued function, $V(\cdot)$, which associates every subset $G$ of $N$ a real number, $V(G)$ [17]. For each coalition, $G$, the number $V(G)$ represents the total payoff to be divided among the members of $G$, i.e.,

$$V(G) = \sum_{\forall x \in G} v(x), \tag{13}$$

where $v(x)$ represents the value allocated to player $x$.

Here, $V(G)$ is called the value of the coalition, $G$. Players can form other coalitions to obtain different values. We say that a stable coalition is formed when players have no incentive to deviate from joining the coalition. Specifically, a coalition, $G$, is stable if we cannot find a better coalition, $G'$, $G' \subseteq G$, with respect to $V(\cdot)$. This implies that

$$\sum_{x \in G} v(x) \geq V(G') \quad \forall G' \subseteq G. \tag{14}$$

In other words, if a coalition is unstable, it is possible for a subset of players to deviate such that each deviating player can obtain a larger value than they do staying put. In game theory literature, the above definition of stability is called the *core* of the cooperative game [17]. In our context, it is undesirable for peers to deviate after joining as that would disrupt the structure of the P2P network and, in turn, adversely affect the streaming quality.

With the above definitions, we devise a cooperative game, called the *peer selection game*, to model the peer selection process. The players are a parent $p$ and a set of children, $c_1, c_2, \ldots, c_n$. We denote the set of all players as, $G_a$, i.e.,

$$G_a = \{p, c_1, c_2, \ldots, c_n\}. \tag{15}$$

The players can freely form other coalitions, $G$, among themselves, where $G \subseteq G_a$. In general, different coalitions lead to different values. The function $V(G)$ should satisfy the following conditions:

$$V(G) = 0 \quad \text{if } p \notin G, \tag{16}$$

$$V(G) \leq V(G') \quad \text{if } G \subseteq G', \tag{17}$$

$$V(G_1 \cup c_i) - V(G_1) \neq V(G_2 \cup c_i) - V(G_2). \tag{18}$$

Condition (16) dictates that the parent, $p$, is a necessary member in any coalitions that generate nonzero values. In other words, $p$ is the veto player of the game. This is a reflection of the reality, where downstream peers depend on their parent for media packets. Without the participation of $p$, a coalition does not bring any value to the members.

Condition (17) indicates that when comparing two coalitions, $G$ and $G'$, the coalition with more members always generates a value no smaller than the other does. This property precisely models a practical scenario, where a parent having a larger number of children is more important because if such a parent departs, a large number of other peers will be disconnected. Thus, the system should attach a higher value to such a coalition.

Condition (18) means that, in general, the same peer, $c_i$, brings different marginal utilities to different coalitions. The discrepancy is attributed to the heterogeneous nature of P2P media streaming. For instance, the presence of $c_i$ would be more significant if the coalition contains only a few children. On the other hand, $c_i$ does not create much value if it joins another coalition already having many children.

We require the value function $V(G)$ to satisfy all the three conditions discussed above. However, the precise definitions depend on the specific characteristics of the application. We will describe a novel value function in Section 3.1.

In the peer selection game, the formation of a coalition $G$, would create an aggregate value, represented by $V(G) = \sum_{\forall x \in G} v(x)$, where $v(x)$ represents the utility allocated to player $x$. We assume that each player would like to maximize its share of utility, i.e., player $x$ is interested in maximizing $v(x)$. This is reasonable because each peer $x$ is more concerned with its own performance in terms of $v(x)$. On the other hand, $V(G)$ is a measure of the value of coalition $G$ in the P2P network.

We also take the participating cost of peers into consideration. Specifically, player $x$ incurs some cost to be a member of a coalition. We denote the amount of player $x$'s coalitional effort as $e(x)$. This can be interpreted as the amount of outgoing bandwidth and other resources consumed. We define the utility of player $x$ as the difference between the share of value obtained from the coalition $v(x)$ and the amount of effort contributed to the coalition $e(x)$. That is, utility is defined as

$$u(x) = v(x) - e(x). \tag{19}$$

Moreover, we assume that $e(x)$ depends on the number of peers in the coalition, i.e.,

$$e(x) = \begin{cases} (|G| - 1)e & x = p \\ e & x \in G \setminus \{p\}, \end{cases} \tag{20}$$

where $e$ is a non-negative constant.

It is clear that if player $x$ does not join any coalition, its utility is zero, i.e., $u(x) = 0$ if $x \notin G$. This implies that a rational player will only join a coalition providing non-negative utility, $u(x) \geq 0$ if $x \in G$. This is called the *incentive compatibility* constraint:

$$u(x) \geq 0 \quad \text{if } x \in G. \tag{21}$$

We denote $G_a$ as the set of players and analyze the peer selection game as $G_a$ increases.

*Case 1.* $G_a = \{p\}$. This is the baseline case where the parent is the sole player. There is only one possible coalition, $G_1 = \{p\}$. The player obtains all the value created by the coalition, which is given by

$$V(G_1) = v(p). \tag{22}$$

Since $p$ has no downstream peer, its effort is zero, i.e., $e(p) = 0$. The utility of $p$ is $u(p) = v(p)$.

*Case 2.* $G_a = \{p, c_1\}$. The set of players include the parent and one potential child, i.e., $P = \{p, c_1\}$. If $p$ accepts $c_i$ as its child, they form a coalition, $G_2 = \{p, c_1\}$. The value created by the coalition is to be distributed between the two players, i.e.,

$$V(G_2) = v(p) + v(c_1). \tag{23}$$

The value $V(G_2)$ needs to be distributed judiciously in order to make $G_2$ a stable coalition such that neither $p$ nor $c_1$ has no incentive to leave. This requires the following conditions to be satisfied:

$$v(p) - e \geq V(G_1), \tag{24}$$

$$v(c_1) - e \geq 0. \tag{25}$$

Condition (24) suggests that $p$ should receive a utility larger than the value created by acting alone. Condition (25) requires that the share of value allocated to $c_1$ should be at least the amount of its contributed effort.

In other words, the share of value allocated $c_1$, denoted by $v(c_1)$, should be

$$e \leq v(c_1) \leq V(G_2) - V(G_1)e. \tag{26}$$

*Case 3.* $G_a = \{p, c_1, c_2\}$. The set of players now includes $p$ and two potential children, i.e., $P = \{p, c_1, c_2\}$. If the parent accepts both peers, they form a larger coalition, $G_3$ and create a value of $V(G_3)$. This is to be distributed among the three players:

$$V(G_3) = v(p) + v(c_1) + v(c_2). \tag{27}$$

We should ensure that $G_3$ is a stable coalition, where the parent and the two children have no incentive to leave. This requires the following conditions to be satisfied:

$$v(p) - 2e \geq V(G_1), \tag{28}$$

$$v(c_1) - e \geq 0, \tag{29}$$

$$v(c_2) - e \geq 0, \tag{30}$$

$$v(p) + v(c_1) \geq V(\{p, c_1\}), \tag{31}$$

$$v(p) + v(c_2) \geq V(\{p, c_2\}). \tag{32}$$

Condition (28) ensures that the parent would not drop the two children. Conditions (29) and (30) lead to nonnegative utilities for $c_1$ and $c_2$, respectively. In other words, these two conditions are the incentive compatibility constraint in (21). The last two conditions, on the other hand, cause dropping one of the two children an undesirable move. We can simplify the conditions as follows:

$$v(c_1) \leq V(G_3) - V(\{p, c_2\}), \tag{33}$$

$$v(c_2) \leq V(G_3) - V(\{p, c_1\}), \tag{34}$$

$$v(c_1) + v(c_2) \leq V(G_3) - V(G_1) - 2e, \tag{35}$$

$$v(c_1), v(c_2) \geq e. \tag{36}$$

*Case n.* $G_a = \{p, c_1, \ldots, c_{n-1}\}$. This is the general scenario where the parent is encountered with $(n-1)$ potential children. If they form a single coalition of size $n$, this creates a value of $V(G_n)$, which is to be distributed among the members, i.e.,

$$V(G_n) = v(p) + \sum_{c_i \in G_n} v(c_i). \tag{37}$$

For $G_n$ to be stable, peers should have no incentive to leave the coalition individually or as a group. Similar to previous cases, we obtain the following conditions:

$$v(c_r) \leq V(G_n) - V(G_n \setminus \{c_r\}) \quad \forall c_r, \tag{38}$$

$$\sum_{\forall c_i \in P} v(c_i) \leq V(G_n) - V(G_1) - (n-1)e, \tag{39}$$

$$v(c_r) \geq e \quad \forall c_r. \tag{40}$$

The term "$V(G_n) - V(G_n \setminus \{c_r\})$" is called the marginal utility of $c_r$. It is the additional amount of value created by $c_r$ to the original coalition. Since $p$'s effort is increased by $e$ if $c_r$ is accepted as its child, the share of value allocated to $c_r$ is

$$v(c_r) = V(G_n) - V(G_n \setminus \{c_r\})e. \tag{41}$$

### 3.1 A Specific Value Function

We propose the following specific value function for our peer selection game:

$$V(G) = \begin{cases} \log\left(1 + \sum_{\forall i \neq p} \frac{1}{b_i}\right) & p \in G, \\ 0 & \text{otherwise}. \end{cases} \tag{42}$$

Without loss of generality, the value function is zero when the parent is the sole coalition member, i.e., $V(G_1) = 0$. This is an increasing function in coalition size. In other words, a new peer always brings additional value to an existing coalition. Furthermore, a peer may create different values to different coalitions. Therefore, the value function satisfies Conditions (16), (17), and (18).

Besides the above characteristics, the value function can also differentiate peers according to their outgoing bandwidth values. For the same coalition, $G$, peer $x$ would receive a larger share of the value than peer $y$ if $b_x < b_y$. The reason for that arrangement would become evident when we discuss the proposed peer selection protocol in Section 4. We illustrate the details with the following numerical example.

Consider two coalitions: $G_X$ and $G_Y$, where $G_X = \{p_x, c_1, c_2\}$ and $G_Y = \{p_y, c_3, c_4, c_5\}$. A peer $c_6$ would like to join one of the two coalitions. We take $e = 0.01$ and the outgoing bandwidth of the peers are listed as follows:

| $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ |
|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 3 | 2 |

It is easy to see that $V(G_X) = 0.92$ and $V(G_Y) = 0.85$. If $c_6$ joins the coalition $G_X$, we have $V(G'_X) = 1.10$ and its share of value is: $V(G'_X) - V(G_X) - e = 0.17$. On the other hand, $c_6$ joining coalition $G_Y$ would result in $V(G'_Y) = 1.04$, and its share of value is $V(G'_Y) - V(G_Y) - e = 0.18$. Therefore, $c_6$ joins $G_Y$ and $v(c_6) = 0.18$. The peer's share of value, i.e., $v(x)$, in the coalition is then used by the parent in determining the amount of bandwidth allocation, which will be discussed in Section 4.

## 4 Proposed Protocol

The proposed peer selection game addresses how a peer selects the best parent with respect to its share of value obtained. Furthermore, the coalition formed is stable in that each and every member has no incentive to leave unilaterally or as a group. Specifically, our proposed schemes are fully distributed so that the participating peers are completely autonomous in deciding, which children to be admitted or which parents to join. Second, our proposed schemes are of low complexity because participating peers do not need to keep heavy weighted state information and carry out extensive message exchanges. Third, our proposed schemes are robust to peer dynamics in that even if a certain peer, possibly acting as a parent, involuntarily departs the system (e.g., due to unexpected machine failures), its associated children can get new connections with other parents. This is demonstrated by our simulation results below.

As discussed in Section 2, we should allow peers to have multiple upstream peers to improve resilience against peer dynamics. Unlike existing approaches, we proposed to let the number of upstream peers depend on the peer's outgoing bandwidth. Specifically, peer $x$ joins the P2P media streaming network by obtaining a list of $m$ candidate parents from the server. Here, we assume that similar to the case of a BitTorrent system [4], such a list can be obtained from a number of "trackers," which can be reached by a well-known address. Then, peer $x$ contacts these candidate parents. Each parent $y$ replies with a bandwidth allocation, $b(x, y)$, which is made proportional to peer $x$'s share of value, i.e.,

$$b(x, y) = \alpha \times v(c_x), \tag{43}$$

where $\alpha$ is a performance parameter, called the *allocation factor*.

It should be noted that $b(x, y)$ is normalized with respect $r$. If peer $x$ cannot obtain the necessary bandwidth from the best candidate, peer $y$, alone, i.e., $b(x, y) < 1$, peer $x$ accepts the second best candidate as its upstream peer. This process continues until the aggregate bandwidth allocated by upstream peers can support the media rate. We detail the peer selection protocols for parent peers and child peers in Algorithms 1 and 2, respectively. The following numerical example illustrates the proposed peer selection process.

**Algorithm 1.** The proposed peer selection protocol (parent).

```
$\alpha$: allocation factor;
Parent $p_y$
Upon receiving a request from a potential, $c_x$, do:
Calculate: $v(c_x) = V(G_Y \cup c_x) - V(G_Y) - e$;
if $v(c_x) \geq e$ then
    reply with the bandwidth allocation, $b_{x,y} = \alpha \times v(c_x)$;
else
    reply with the bandwidth allocation, $b_{x,y} = 0$;
end if
```

**Algorithm 2.** The proposed peer selection protocol (child).

```
$m$: number of candidate parents (obtained from server);
Child $c_x$
Send requests to the m candidate parents;
Each candidate parent $y$ replies with its allocation, $b_{x,y}$;
Denote: $B' = \{b_{x,y}\}, \forall y$;
$B = \{\emptyset\}$;
$b = 0$;
while $b < 1$ do
    Select the largest allocation, $b_{x,y'}$, from $B$;
    $B = B \cup y'$;
    $b = b + b_{x,y'}$;
end while
Cancel the allocation for those parent(s) not listed in $B$;
Confirm the allocation for those parent(s) listed in $B$;
```

We take $\alpha = 1.5$ and $m = 5$. In other words, a joining peer is given five candidate parents. For simplicity, we assume these five candidates have no downstream peer such that they have the same bandwidth allocation to a particular peer. Consider three joining peers: $c_1$, $c_2$, and $c_5$. Since $b_1 = 1$, we have $v(c_1) = 0.68$, and its allocated bandwidth is $\alpha \times v(c_1) = 1.02 > 1$. Thus, $c_1$ has one upstream peer. For $c_2$, its outgoing bandwidth is $b_2 = 2$ and $v(c_2) = 0.40$. Each candidate parent allocates $\alpha \times v(c_2) = 0.59$. This implies that $c_2$ should select two candidates as its upstream peers. Finally, $c_5$'s outgoing bandwidth is $b_5 = 3$, and $v(c_5)$ is 0.28. As such, $c_5$ accepts three upstream peers.

From the above illustration, we can see that the number of upstream peers accepted by a particular peer depends on the peer's outgoing bandwidth, and the allocation factor, $\alpha$. The value function, defined in (42), implies that a peer with a larger contribution would be provided with more upstream peers, and vice versa. This causes the share of value decrease with increasing outgoing bandwidth. This is because we would like to provide more parents to peers with larger outgoing bandwidth. With more parents, they are less likely to be affected by peer dynamics. On the other hand, peers having smaller outgoing bandwidth accept fewer parents, which reduce the average number of links per peer. Although they are more vulnerable to peer dynamics, we expect this effect is limited as they have only a few children. Simulation results in Section 5 suggest that we can improve delivery ratio with comparable average number of links per peer. We note that the proposed peer selection process also serves as an incentive to contribute.

Indeed, a peer with a larger outgoing bandwidth would in fact get a larger number of parents (i.e., upstream peers), with each of which providing a smaller "share" of coalition value, i.e., a small value of incoming bandwidth. This arrangement is to enhance the resilience and robustness of such high-contribution peer in that even if one of its parent departs the system, it will still have plenty of other parents to supply data to it. On the other hand, for a peer with a smaller outgoing bandwidth, it will get a larger "share" of the coalition value from each parent, but then, it will have much fewer parents associated with it. Thus, such a peer will not be as resilient to peer dynamics as compared to those high-contribution peers.

We denote our proposed peer selection protocol as, $\mathrm{Game}(\alpha)$. Table 1 compares the proposed protocol with other existing approaches. We can see that existing approaches assign upstream peers without regard to a peer's outgoing bandwidth. On the other hand, the proposed protocol allows the number of upstream peers to be dependent on the peer's outgoing bandwidth. Furthermore, peers have no incentive to relocate themselves for better performance. This leads to a

TABLE 1
Comparison of Different P2P Media Streaming Approaches

| | Number of upstream peers (parents) | Number of downstream peers (children) | Average number of links per peer |
|---|---|---|---|
| Tree(1) | 1 | $\lfloor \frac{b_x}{r} \rfloor$ | $O(1)$ |
| Tree($k$) | $k$ | $\lfloor \frac{b_x}{r/k} \rfloor$ | $O(k)$ |
| DAG($i$,$j$) | $i$ | $j$ | $O(i)$ |
| Unstruct($n$) | $n$ | $n$ | $O(n)$ |
| Game($\alpha$) | depends on $b_x$ and $\alpha$ | depends on $\alpha$ | $O(\alpha)$ |

more efficient allocation of P2P link and, hence, a smaller amount of communication overhead.

In our protocols, any peer can choose to be a parent. Specifically, when the system starts, there will be an initial set of participants where they connect to the server directly, and yet, each of them acts as a "null" parent in the sense that there is not any child connecting to it. Subsequently, a new peer joining the system could also opt to connect to the server directly *under the condition that* it cannot get a higher $v(x)$ value, compared to its $V(x)$ value as a "null" parent, if it tries to associate with one of the existing peers. Depending on the distribution of the incoming and outgoing bandwidth values of the participating peers, some newly joined peers will eventually find it more beneficial in associating with an existing peer as a child rather than to act alone. As the system evolves, a hierarchy will be formed.

## 5 Performance Evaluation

In this section, we evaluate the performance of the proposed game-theoretic peer selection protocol. We implemented four existing approaches for performance comparison:

- *Random.* We have implemented a totally random peer selection approach (similar in essence to the probabilistic peer selection schemes used in contemporary P2P systems such as BitTorrent [4]) as a baseline approach in our comparison.
- *Tree(1).* Peers are organized in a single tree. Each peer has one parent, and the number of children depends on the peer's outgoing bandwidth, as shown in Table 1.
- *Tree(4).* Peers are organized in 4 independent trees. This allows each peer to have four parents, while the number of children is also dependent on the peer's outgoing bandwidth, as shown in Table 1.
- *DAG(3, 15).* Peers are organized in a DAG. In particular, we let each peer accept three parents and 15 children, i.e., $i = 3$, and $j = 15$, as shown in Table 1. The same settings were also used in [16].
- *Unstruct(5).* Peers are organized in an unstructured overlay, i.e., a random graph. However, this is different from the Random approach mentioned above in that the system imposes an upper bound on peer connectivity. Specifically, we set $n = 5$ such that each peer is assigned with five neighbors. Since there are up to 3,000 peers among the simulation scenarios, each peer should have at least $0.5139 \times \log(3{,}000)$ neighbors for connectedness with high probability, i.e., $n = 5$.
- *Game(1.5).* Peers are organized in a generalized DAG using the proposed peer selection protocol detailed in Algorithms 1 and 2. The allocation factor, $\alpha$, is taken to be 1.5 and the cost constant, $e$, is taken to be 0.01. We will investigate the effect of $\alpha$ in Section 5.4.

The performance of P2P media streaming is quantified with five performance metrics:

1. *delivery ratio*, the ratio between the number of received packets to that of generated packets,
2. *number of joins*, the number of joining peers, which includes new peers and those existing peers who are forced to rejoin due to peer dynamics,
3. *number of new links*, the number of links created due to peer dynamics,
4. *average packet delay*, and
5. *average number of links per peer*.

We use the GT-ITM [10] network topology generator to generate physical topology based on the transit-stub scheme. The network consists of one transit domain with 50 nodes. The mean link delay is set to 30 ms, which represents the backbone network. Each node in the transit domain is connected to five stub domains, each of which has 20 nodes. The mean link delay is set to 3 ms, representing the edge network. As such, there are 5,000 edge nodes in total. We randomly select some edge nodes to act as peers. Unless otherwise specified, the outgoing bandwidth of peers is randomly chosen between 500 Kbps and 1,500 Kbps. Table 2 lists the values of other simulation parameters.

### 5.1 Effect of Turnover Rate

We varied the turnover rate between 0 percent and 50 percent, and the results are shown in Fig. 2. The turnover rate is defined as the percentage of peers leave-and-rejoin throughout the media streaming session. For example, if the turnover rate is at 20 percent, there are 200 leave-and-join operations. This measures how the performance of the media streaming network in the presence of peer dynamics.

Figs. 2a and 2b shows the delivery ratio of peers when the turnover rate is 0-25 percent and 25-50 percent,

TABLE 2
Simulation Parameters

| Definition | Default | Range |
|---|---|---|
| Number of peers | 1000 | 500–3000 |
| Outgoing bandwidth of server | 3000kbps | |
| Outgoing bandwidth of peers (min) | 500kbps | |
| Outgoing bandwidth of peers (max) | 1500kbps | 1000–3000kbps |
| Media rate | 500kbps | |
| Turnover rate | 20% | 0–50% |
| Allocation factor ($\alpha$) | 1.5 | 1.2–2.0 |
| Session duration | 30min | |

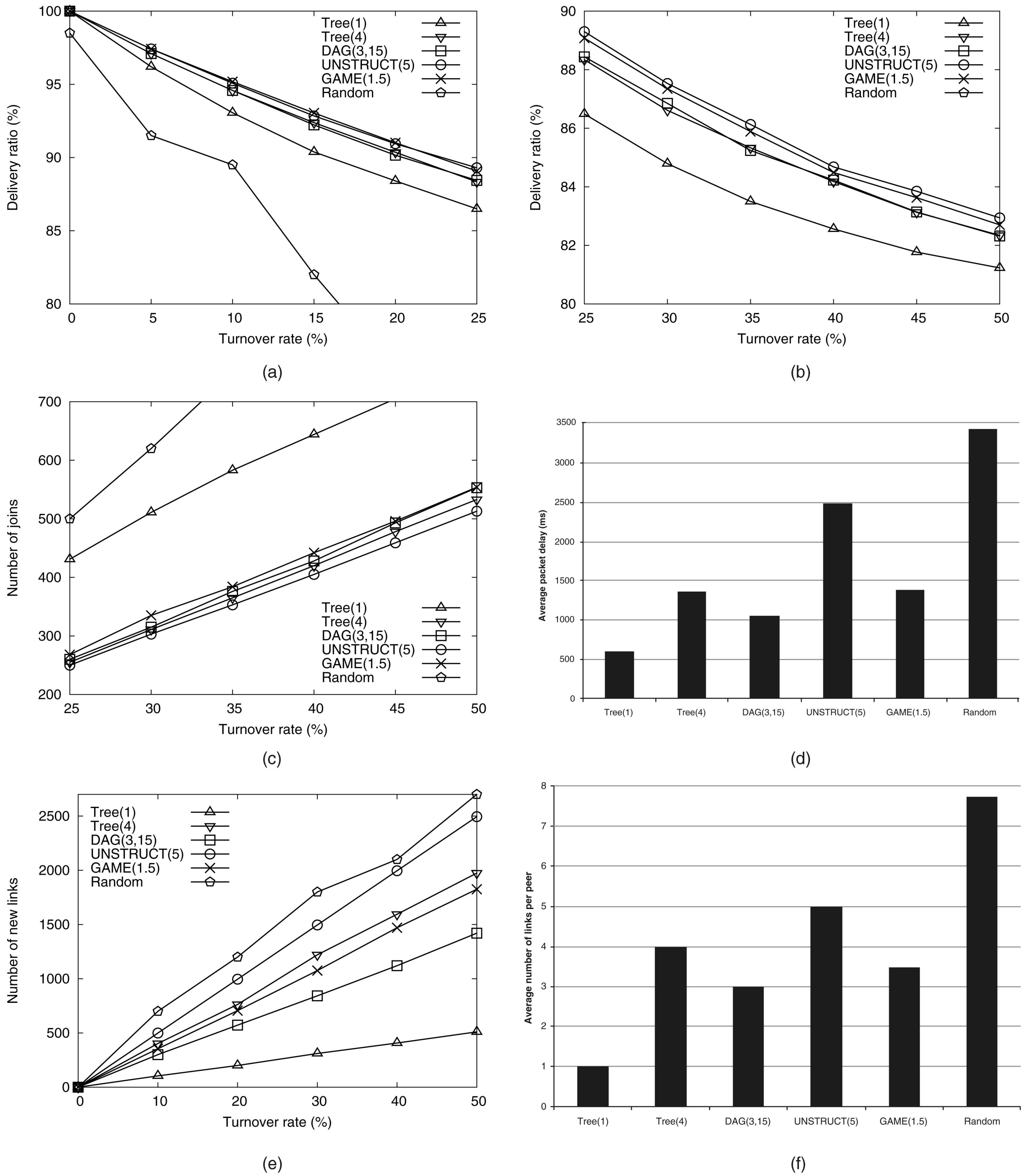
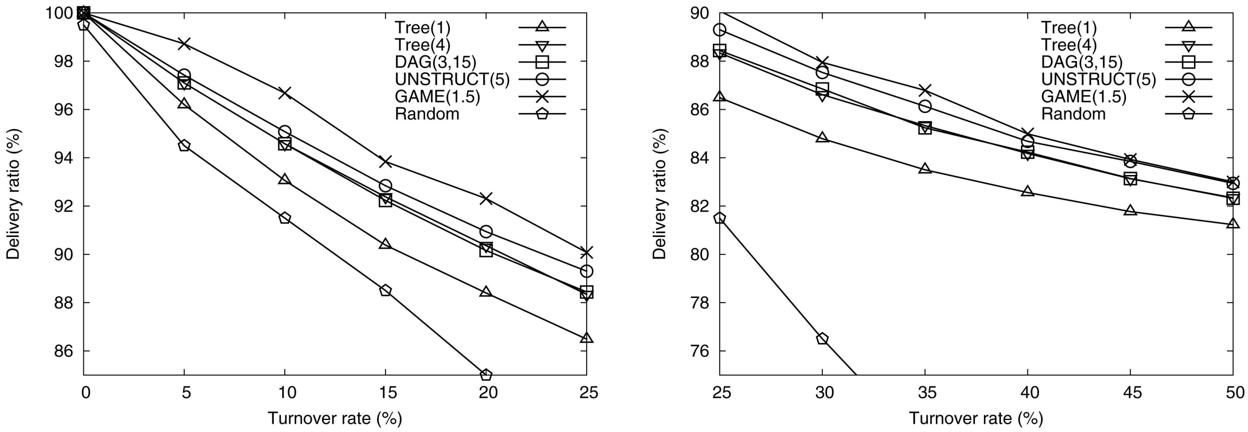


Fig. 2. Effect of turnover rate (random join and leave). (a) Delivery ratio. (b) Delivery ratio. (c) Number of joins. (d) Average packet delay. (e) Number of new links. (f) Average number of links per peer.

respectively. It is evident that the single tree approach, Tree(1), results in the worst delivery ratio. As we discussed in Section 2, each peer depends on its sole parent for media packets. If that parent leaves the session, the peer cannot receive any media packets until a new parent is found. This inevitably lowers the delivery ratio. On the other hand, the two structured approaches (Tree(4) and DAG(3,15)) gives comparable performance throughout the whole simulation range. This may be attributed to the fact that the DAG approach is a generalization of multiple trees, only without the need to maintain more than one structure. The proposed approach, which is based on the peer selection game described in Section 3, shows some improvements over other structured approaches. Furthermore, Game(1.5) is in par with Unstruct(5) at low turnover rates, i.e., between 0 percent and 25 percent. This indicates that the proposed peer selection protocol is effective in mitigating the performance degradation due to peer dynamics. Since the



(a)

(b)

Fig. 3. Effect of turnover rate (nonrandom join and leave). (a) Delivery ratio. (b) Delivery ratio.

unstructured approach organizes peers in a random graph, the inherent random arrangement of peers means that the performance is least affected by peer dynamics.

We turn our attention to the number of joins throughout the session. Notice that all approaches, except Tree(1), give almost identical results at low turnover rates. We only show the number of joins at turnover rates between 25 percent and 50 percent, as shown in Fig. 2c. We observe that Tree(1) generates the largest number of join operations among all the approaches, which is the major reason for its worst delivery ratio. This also follows that Unstruct(5) results in the smallest number of joins. As peers are interconnected to form a random graph for media streaming, a peer is forced to rejoin only when all of its neighbors fail, which is unlikely to occur for $n = 5$. Of the structured approaches, the proposed protocol results in the largest number of joins. This is because the peer selection process gives more parents to peers having higher outgoing bandwidth, which causes the other peers more susceptible to peer dynamics. Therefore, there is slightly a larger number of joins.

We measured the average delay of received media packets and the result is shown in Fig. 2d. Tree(1) builds a single tree, which is the most efficient and achieves the least average packet delay. On the other extreme, Unstruct(5) builds a random graph, which significantly increases the average packet delay because of random arrangement. The proposed peer selection protocol manages to achieve comparable performance with the other two structured approaches. Unlike Tree(1), media packets can achieve at peers from different paths for Tree(4), DAG(3,15), and Game(1.5). As such, peers experience longer average packet delay. Actually, the average packet delay generally increases with the number of possible paths.

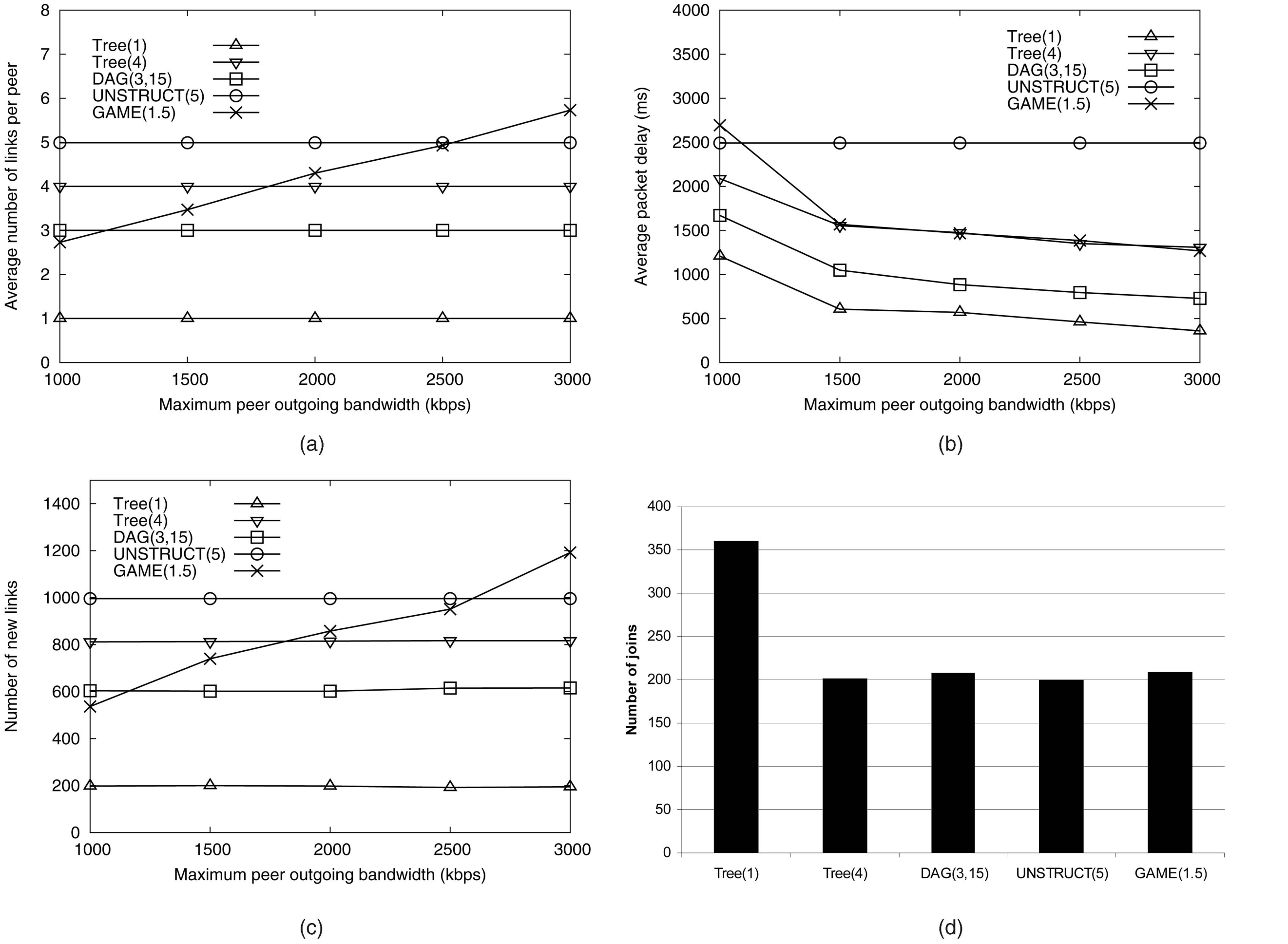
Fig. 2d shows the number of new links when the turnover rate ranges from 0 percent to 50 percent. For all approaches, the number of new links increases almost linearly with turnover rate. This is because the number of new links created is closely related to turnover. As the turnover rate increases, there are more peers joining the P2P overlay, which creates more new links to existing peers. Besides that, existing peers establish new links to compensate for those lost as a result of peer dynamics. Therefore, we have the almost linear trend. However, their rates of increase differ a lot, with Tree(1) having the lowest rate and Unstruct(5) having the largest rate. The ordering can be explained by the average number of links per peer, see Fig. 2f. Of the existing approaches, their simulation results match closely with the values in Table 1. For Game(1.5), the average number of links per peer is 3.47, which slightly larger than DAG(3,15) but smaller than Tree(4). The number of links per peer can be considered as a measure of overheads incurred in P2P media streaming. The proposed peer selection protocol strategically establishes P2P links such that peers are assigned with a number of parents depending on their outgoing bandwidth. On the average, peers in the proposed approach have comparable number of links with those in existing structured approaches.

We have assumed that the join-and-leave peers are randomly chosen from the entire population. However, it is reasonable to expect that peers with low contribution are more likely to leave the session. They may be considered as users choosing from different available channels before settling at a particular one. Fig. 3 shows the delivery ratio when the join-and-leave peers are selected among peers with the smallest outgoing bandwidth. Since the four existing approaches do not take peer contribution into consideration, their performance remains the same. Nevertheless, we also plotted their results for comparison. On the other hand, the proposed approach shows consistent improvement across the entire range of turnover rates. This indicates that the proposed peer selection game provides more parents to peers with higher outgoing bandwidth, leading to the overall improvement in delivery ratio. At increasing turnover rates, the performance of Game(1.5) gradually decreases and approaches that of $\mathrm{Unstruct}(n)$. This confirms that peers contributing larger outgoing bandwidth are more important to the overall performance.

To summarize, the proposed peer selection protocol achieves better delivery ratio than existing structured approaches using comparable number of links per peer. At low turnover rates, the performance is in par with the unstructured approach. It shows more improvement if peer dynamics is focused on peers with low contribution. As for average packet delay, the proposed approach is also similar to other structured counterparts. This suggests that the proposed peer selection game is effective in forming stable coalitions resulting in overall performance improvement.

## 5.2 Effect of Outgoing Bandwidth of Peers

Fig. 4 shows the effect of outgoing bandwidth of peers. The minimum outgoing bandwidth was kept at 500 Kbps, while



(a) (b) (c) (d)

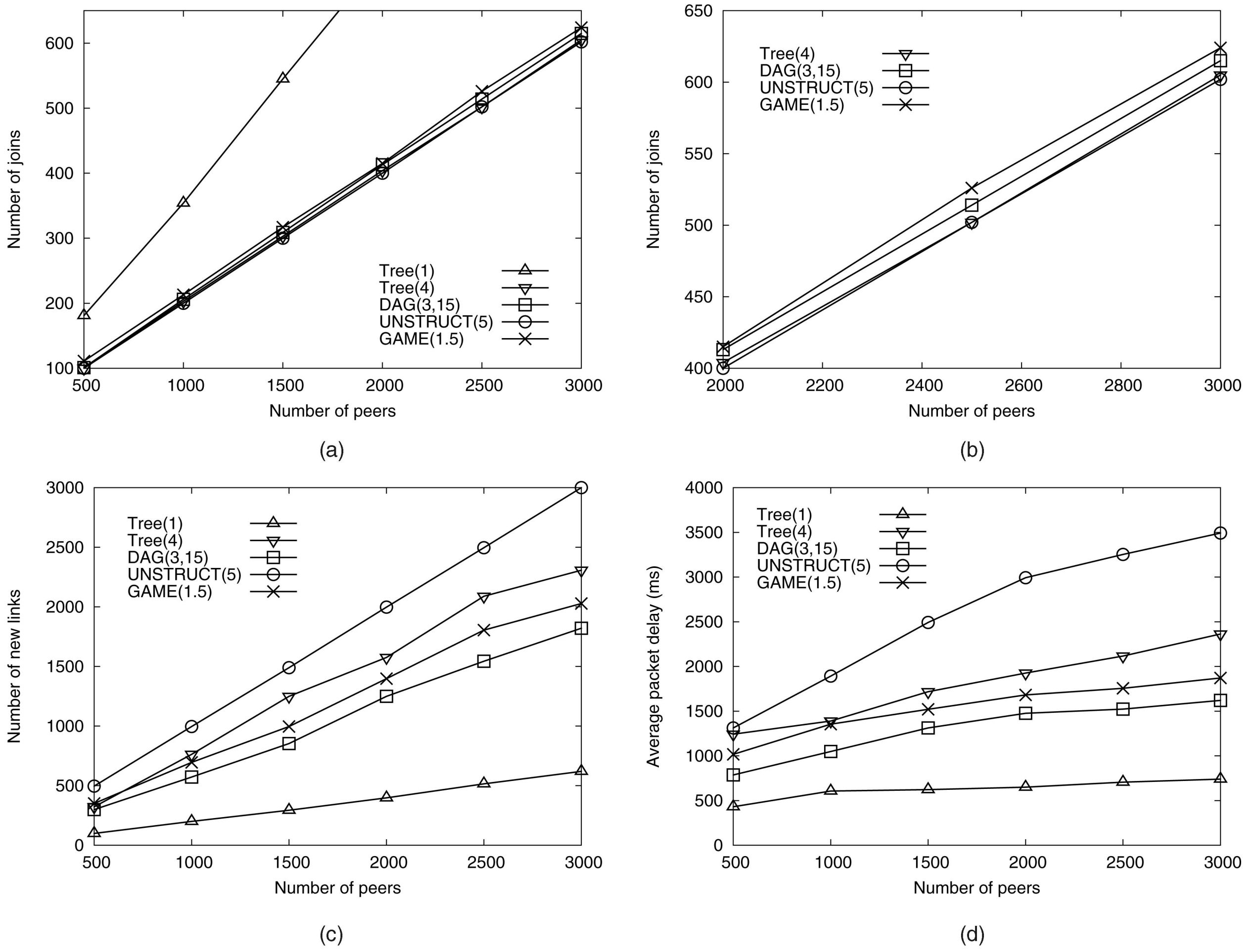
Fig. 4. Effect of outgoing bandwidth of peers. (a) Average number of links per peer. (b) Average packet delay. (c) Number of new links. (d) Number of joins.

the maximum value ranges from 1,000 Kbps to 3,000 Kbps. This studies the impact of different peer contributions on various approaches.

Since existing approaches (Tree(1), Tree(4), DAG(3,15), and Unstruct(5)) do not consider peer contribution in constructing the structures, the change in peer outgoing bandwidth has no effect on the average number of links per peer, see Fig. 4a. For the proposed approach, increasing peer outgoing bandwidth leads to the rise in the average number of links per peer, even the allocation factor remains the same. This suggests that the average overheads experienced by a peer increase with its contribution. This is due to the design of the proposed peer selection game. The amount of bandwidth allocation to peer $x$, $v(x)$, decreases with its outgoing bandwidth, $b_x$. Therefore, the larger the $b_x$, the more is the number of its upstream peers. As such, we have witnessed the upward trend. We believe that this is a reasonable arrangement because peers who can afford to contribute more are expected to incur increased maintenance overheads.

The change in peer outgoing bandwidth also affects the average packet delay, see Fig. 4b. Of all structured approaches, average packet delay gradually decreases with increasing outgoing bandwidth. With reference to Table 1, the number of downstream peers for Tree(1) and Tree(4) are directly related to outgoing bandwidth. If peers can accept more children, the height of the tree structures would be shorter, which reduces the average packet delay. Although DAG(3,15) limits the number of downstream peers to 15, most peers cannot accept so many children, i.e., the constraint is not always active. When we increase the outgoing bandwidth, some peers can still accommodate more children. The effect is that the DAG structure can be shortened, reducing the average packet delay. For the proposed approach, we can observe from Algorithm 1 that the amount of bandwidth allocation depends on the parent's outgoing bandwidth and the allocation factor, $\alpha$. Therefore, the average packet delay of Game(1.5) also shows a deceasing trend. The above discussions, however, do not apply to $\mathrm{Unstruct}(n)$. Since the number of neighbors are fixed at five, its average packet delay remains unchanged throughout the entire range.

Fig. 4c shows the number of new links created as we varied the maximum peer outgoing bandwidth between 1,000 Kbps and 3,000 Kbps. All existing approaches are not affected because the number of links per peer also remains unchanged. Similarly, the increasing trend in Game(1.5) is primarily due to the increase in the average number of links per peer. This explanation is confirmed by the number of joins, which is not affected by peer outgoing bandwidth. For brevity, we use a bar chart as shown in Fig. 4d. Although Game(1.5) achieves similar number of joins with other



(a)

(b)

(c)

(d)

Fig. 5. Effect of peer population size. (a) Number of joins. (b) Number of joins. (c) Number of new links. (d) Average packet delay.

approaches, it has increasing number of links per peer. This leads to the rising trend in the number of new links.

To summarize, all structured P2P media streaming approaches give decreasingly small average packet delay when peers contribute more resources. However, the proposed peer selection protocol shows a unique characteristic: the average number of links per peer increases with outgoing bandwidth. We believe that this arrangement is justifiable because a peer contributing more resources is more willing to maintain more links.

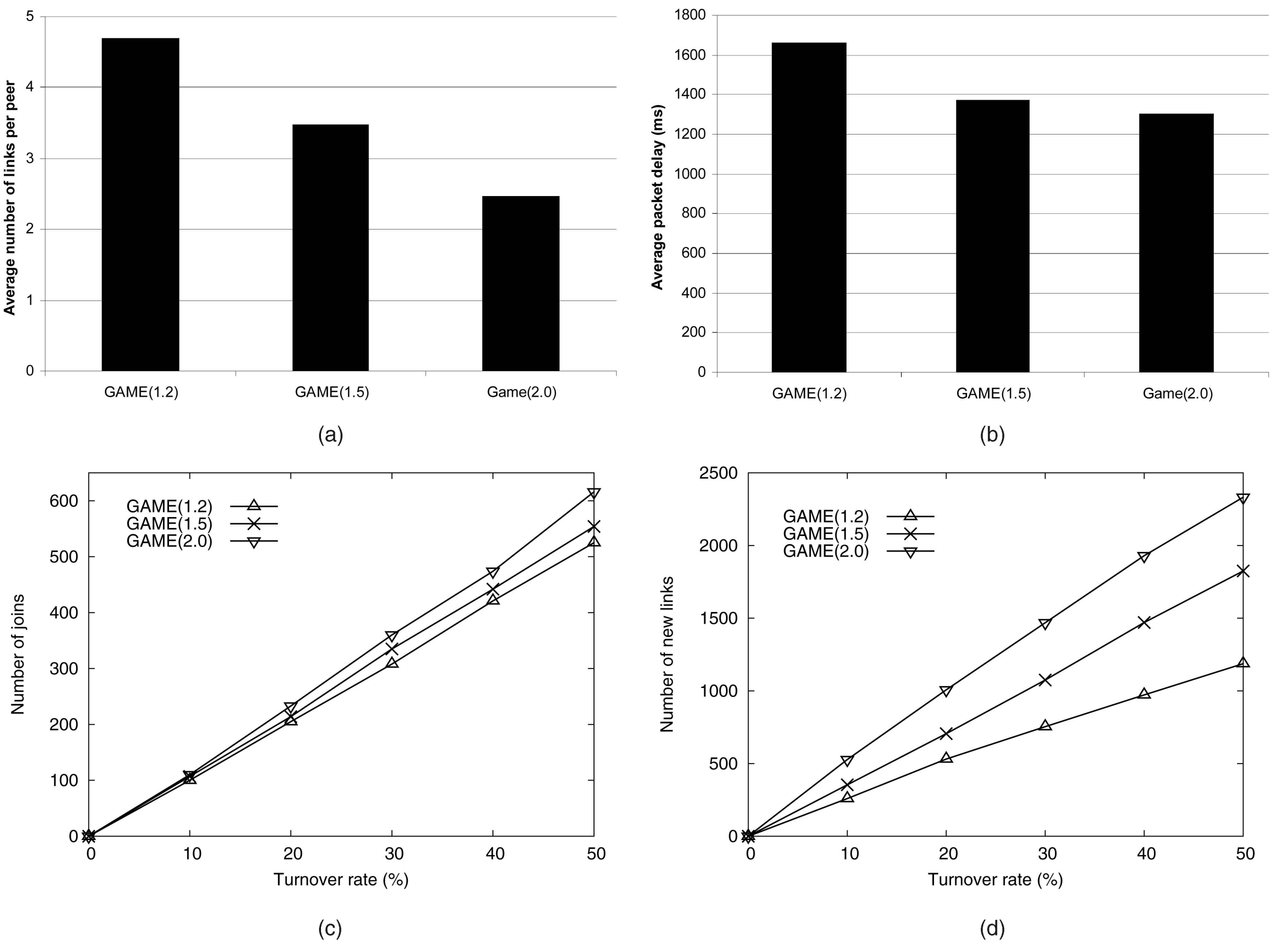
### 5.3 Effect of Peer Population Size

This set of simulation studies the effect of number of peer population size on various performance metrics, see Fig. 5. Specifically, we varied the number of peers between 500 and 3,000 and measured the number of joins, number of new links, and average packet delay.

Although the turnover rate was kept at 20 percent, the larger peer population size leads to the corresponding increase in the number of join-and-leave peers. This translates to the rising trend in number of joins, as shown in Fig. 5a. Except Tree(1), all approaches show indistinguishable performance. As discussed previously, the exceptional large number of joins in Tree(1) is due to its arrangement where peers depend on their sole parent for all media packets. The other approaches provide multiple paths for packet delivery. However, the performance difference slightly increases at large peer population size. Fig. 5b is the magnified view of Fig. 5a for the range 2,000-3,000. We can see that the proposed peer selection protocol gives marginally more number of joins compared with other approaches. This is because of the dynamic peer selection process. Peers with smaller outgoing bandwidth are prone to be more affected by peer dynamics. As such, it is more likely for them to be isolated, which increases the number of joins.

Although the proposed approach requires slightly more number of join operations, its number of new links is comparable to other structured approaches, see Fig. 5c. This is because the number of new links is affected by two factors: number of joining peers and number of links affected by peer dynamics. When a peer leaves the structure, the links to downstream peers are broken. These affected peers are required to establish new links for compensation. Therefore, the proposed approach manages to limit the number of affected peers through the peer selection process.

Fig. 5d shows the variations of average packet delay when the peer population size is between 500 and 3,000. It is intuitive to observe that the average packet delay increases with the number of peers. This is because media packets are required to traverse more overlay links in order to reach all peers. Of the structured approaches, i.e., Tree(1), Tree(4), DAG(3, 15), and Game(1.5), their rates of increase are comparatively low. This suggests that structured approaches scale very well with respect to the number of peers, which explains their overwhelming popularity. For example, a binary tree can accommodate twice the number



(a)

(b)

(c)

(d)

Fig. 6. Effect of allocation factor, $\alpha$. (a) Average number of links per peer. (b) Average packet delay. (c) Number of joins. (d) Number of new links.

of peers by increasing its height by 1. On the other hand, the unstructured approach, $\mathrm{Unstruct}(n)$, is more sensitive to peer population size, especially at the low end. This is because it sacrifices packet delay in favor of resilience. As such, peers in an unstructured based P2P media streaming network are expected to experience a longer startup time and require a larger buffer.

To summarize, comparing with other existing approaches, the proposed peer selection protocol scales equally well as the peer population size increases. Although there are more joins at the same turnover rate, the proposed approach is able to maintain the number of new links to a reasonable level. This indicates that the proposed peer selection game performs well against peer dynamics.

### 5.4 Effect of Allocation Factor, $\alpha$

We have compared the proposed peer selection protocol with other existing approaches under different settings. In previous scenarios, the allocation factor, $\alpha$, is fixed at 1.5. We would like to investigate the effect of $\alpha$ on the performance of the proposed approach. Specifically, we set $\alpha$ to 1.2 and 2.0, represented by Game(1.2) and Game(2.0), respectively. The results are shown in Fig. 6.

With reference to Fig. 6a, we can see that increasing the allocation factor, $\alpha$, would reduce the average number of links per peer. This is because the larger the allocation factor, the more is the bandwidth allocation to a downstream peer. As such, peers are not necessary to have so many parents provided that other parameters are the same. On the average, each peer has a fewer number of links. If the allocation factor is sufficiently large, the proposed peer selection protocol reduces to Tree(1), i.e., each peer has one parent. Similarly, a large $\alpha$ allow peers to experience a smaller average packet delay, as shown in Fig. 6d. The drop in average packet delay is due to the reduction in number of upstream peers. When media packets have more possible paths to reach a peer, this usually involves a larger number of links, resulting in the decreasing trend.

We kept the number of peers at 1,000 and increased the turnover rate up to 50 percent. Figs. 6c and 6d shows the effect of $\alpha$ on the number of joins and number of new links, respectively. Since each peer has more number of parents when $\alpha$ increases, this improves the overall resilience to peer dynamics. As such, Game(1.2) gives the best results for both metrics. In particular, the performance difference gradually increases with the turnover rate. This indicates that we should set a smaller allocation factor if it is expected to have more join-and-leave activities, e.g., at the beginning or at end of a streaming session.

To summarize, reducing the allocation factor, $\alpha$, would produce more links per peer. This provides better resilience against peer dynamics at the cost of larger average packet delay. On the other hand, if $\alpha$ is sufficiently larger, the proposed peer selection protocol reduces to the single tree approach. By adjusting the allocation factor alone, we can control how peers are organized to disseminate media packets.

## 6 Concluding Remarks

P2P media streaming has attracted significant attention because of its promising scalability and resilience characteristics. There are various existing approaches [11], [12], [13], [15], [16], [19], [20], [21], [25], [30], [31] to achieve P2P media streaming on the global scale. We have classified them in two broad categories: structured and unstructured. For the structured category, the direction of media packets is fixed after a particular peer joins the P2P media streaming network. On the other hand, the unstructured category allows peers to exchange media packets with their neighbors in both directions. We further divided the structured category into three approaches: single tree, multiple trees, and DAG, based on how peers are organized. For each existing approach, we analyzed three important quantities: the number of upstream peers, the number of downstream peers, and the average number of links per peer, see Table 1. Unfortunately, we found that existing approaches do not take peer heterogeneity into consideration. We proposed a peer selection protocol, where each peer judiciously selects an appropriate set of parents depending on its outgoing bandwidth. Specifically, we model the peer selection process as a cooperative game such that peers form stable coalitions with highest values. Simulation results have shown that, comparing with existing approaches, the proposed protocol improves delivery ratio in the presence of peer dynamics. One promising avenue of further research is to extend our study to model a wireless media data access scenario [28], [29].

## Acknowledgments

Preliminary versions of portions of this paper were presented at the 28th International Conference on Distributed Computing Systems (ICDCS 2008), Beijing, China, June 2008.

## References

[1] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, and D.D. Yao, "Optimal Peer Selection for P2P Downloading and Streaming," *Proc. IEEE INFOCOM '05,* vol. 3, pp. 1538-1549, Mar. 2005.

[2] S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," *ACM Computing Surveys,* vol. 36, no. 4, pp. 335-371, Dec. 2004.

[3] P. Baccichet, N. Jeonghun, E. Setton, and B. Girod, "Content-Aware P2P Video Streaming with Low Latency," *Proc. IEEE Int'l Conf. Multimedia and Expo (ICME '07),* pp. 400-403, July 2007.

[4] *BitTorrent,* http://www.bittorrent.com/, 2008.

[5] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-Bandwidth Multicast in Cooperative Environments," *Proc. 19th ACM Symp. Operating Systems Principles (SOSP '03),* pp. 298-313, Oct. 2003.

[6] Y. Chu, S. Rao, S. Seshan, and H. Zhang, "Enabling Conferencing Applications on the Internet Using an Overlay Muilticast Architecture," *ACM SIGCOMM Computer Comm. Rev.,* vol. 31, no. 4, pp. 55-67, Aug. 2001.

[7] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming Live Media over a Peer-to-Peer Network," technical report, Stanford InfoLab, Aug. 2001.

[8] P.B. Godfrey, S. Shenker, and I. Stoica, "Minimizing Churn in Distributed Systems," *ACM SIGCOMM Computer Comm. Rev.,* vol. 36, no. 4, pp. 147-158, Aug. 2006.

[9] V.K. Goyal, "Multiple Description Coding: Compression Meets the Network," *IEEE Signal Processing Magazine,* pp. 74-93, Sept. 2001.

[10] *GT-ITM: Georgia Tech Internetwork Topology Models,* http://www.cc.gatech.edu/projects/gtitm/, 2009.

[11] J. Jannotti, D.K. Gifford, and K.L. Johnson, "Overcast: Reliable Multicasting with an Overlay Network," *Proc. Fourth Usenix Symp. Operating Systems Design and Implementation (OSDI '00),* Oct. 2000.

[12] D. Kostić, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh," *Proc. 19th ACM Symp. Operating Systems Principles (SOSP '03),* pp. 282-297, Oct. 2003.

[13] J. Liang and K. Nahrstedt, "DagStream: Locality Aware and Failure Resilient Peer-to-Peer Streaming," *Proc. SPIE Multimedia Computing and Networking (MMCN '06),* vol. 6071, Jan. 2006.

[14] C. Liang, Y. Guo, and Y. Liu, "Hierarchically Clustered P2P Streaming System," *Proc. Global Telecomm. Conf. (GLOBECOM '07),* pp. 236-241, Nov. 2007.

[15] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A.E. Mohr, "Chainsaw: Eliminating Trees from Overlay Multicast," *Proc. Fourth Int'l Workshop Peer-to-Peer Systems (IPTPS '05),* pp. 127-140, Feb. 2005.

[16] W.T. Ooi, "Dagster: Contributor-Aware End-Host Multicast for Media Streaming in Heterogeneous Environment," *Proc. SPIE Multimedia Computing and Networking (MMCN '05),* vol. 5680, pp. 77-90, Jan. 2005.

[17] M.J. Osborne and A. Rubinstein, *A Course in Game Theory.* MIT Press, July 1994.

[18] J. She and P.-H. Ho, "Cooperative Coded Video Multicast for IPTV Services under EPON-WiMAX Integration," *IEEE Comm. Magazine,* pp. 104-110, Aug. 2008.

[19] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, "The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-Points," *Proc. ACM SIGCOMM '04,* vol. 34, no. 4, pp. 107-120, Aug. 2004.

[20] D.A. Tran, K.A. Hua, and T. Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming," *Proc. IEEE INFOCOM '03,* vol. 2, pp. 1283-1292, Mar./Apr. 2003.

[21] D.A. Tran, K.A. Hua, and T.T. Do, "A Peer-to-Peer Architecture for Media Streaming," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 1, pp. 121-133, Jan. 2004.

[22] Y.-C. Tu, J. Sun, M. Hefeeda, and S. Prabhakar, "An Analytical Study of Peer-to-Peer Media Streaming Systems," *ACM Trans. Multimedia Computing, Comm., and Applications,* vol. 1, no. 4, pp. 354-376, Nov. 2005.

[23] V. Venkataraman, P. Francisy, and J. Calandrino, "Chunkyspread: Multi-Tree Unstructured Peer-to-Peer Multicast," *Proc. Fifth Int'l Workshop Peer-to-Peer Systems (IPTPS '06),* Feb. 2006.

[24] F. Wang, Y. Xiong, and J. Liu, "mTreebone: A Hybrid Tree/Mesh Overlay for Application-Layer Live Video Multicast," *Proc. Int'l Conf. Distributed Computing Systems (ICDCS '07),* June 2007.

[25] J.D. Weisz, S. Kiesler, H. Zhang, Y. Ren, R.E. Kraut, and J.A. Konstan, "Watching Together: Integrating Text Chat with Video," *Proc. ACM SIGCHI '07,* pp. 877-886, Apr./May 2007.

[26] F. Xue and P.R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks," *Proc. Wireless Networks,* vol. 10, no. 2, pp. 169-181, Mar. 2004.

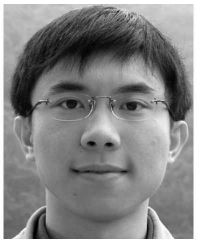
[27] M.K.H. Yeung, "Non-Cooperative Peer-to-Peer Media Streaming: Game Theoretic Analysis and Algorithms," PhD dissertation, Univ. of Hong Kong, Aug. 2007.

[28] M.K.H. Yeung and Y.-K. Kwok, "Wireless Cache Invalidation Schemes with Link Adaptation and Downlink Traffic," *IEEE Trans. Mobile Computing,* vol. 4, no. 1, pp. 68-83, Jan./Feb. 2005.

[29] M.K.H. Yeung and Y.-K. Kwok, "A Game Theoretic Approach to Power Aware Wireless Data Access," *IEEE Trans. Mobile Computing,* vol. 5, no. 8, pp. 1057-1073, Aug. 2006.

[30] M. Zhang, L. Zhao, Y. Tang, J.-G. Luo, and S.-Q. Yang, "Large-Scale Live Media Streaming over Peer-to-Peer Networks through Global Internet," *Proc. Workshop Advances in Peer-to-Peer Multimedia Streaming (P2PMMS '05),* pp. 21-28, Nov. 2005.

[31] X. Zhang, J. Liu, B. Lim, and T.-S.P. Yum, "CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming," *Proc. IEEE INFOCOM '05,* vol. 3, pp. 2102-2111, Mar. 2005.



**Mark Kai Ho Yeung** received the BEng (first class honors), MPhil, and PhD degrees in electrical and electronic engineering from the University of Hong Kong in 2002, 2004, and 2007, respectively. His research interests include wireless cache invalidation, wireless data access, and ad hoc mobile networks. He is a student member of the IEEE and the IEEE Computer Society.

**Yu-Kwong Kwok** received the BSc degree in computer engineering from the University of Hong Kong in 1991 and the MPhil and PhD degrees in computer science from the Hong Kong University of Science and Technology (HKUST) in 1994 and 1997, respectively. He is an associate professor in the Department of Electrical and Computer Engineering, Colorado State University, and on leave of absence from the Department of Electrical and Electronic Engineering, University of Hong Kong (HKU). Before joining HKU in August 1998, he was a visiting scholar for one year in the Parallel Processing Laboratory, School of Electrical and Computer Engineering, Purdue University. He recently served as a visiting associate professor at the Department of Electrical Engineering-Systems, University of Southern California, from August 2004 to July 2005, on his sabbatical leave from HKU. His research interests include distributed computing systems, wireless networking, and mobile computing. He is a senior member of the ACM, the IEEE, and the IEEE Computer Society. He is also a member of the IEEE Communications Society. He received the Outstanding Young Researcher Award from HKU in November 2004.