2010

Incentive Mechanisms in P2P Media Streaming Systems

Xiao Su, *San Jose State University*
Suchreet K. Dhaliwal, *San Jose State University*

Incentive Mechanisms in P2P Media Streaming Systems

Xiao Su, Suchreet K. Dhaliwal
Department of Computer Engineering
San José State University
San José, CA 95192–0180
Email: xsu@email.sjsu.edu, suchreetkaur@gmail.com

Incentive Mechanisms in P2P Media Streaming Systems

Abstract

This paper highlights the need to curb free-riding in P2P media streaming systems and discusses the mechanisms by which this could be accomplished. Free riding, whereby a peer utilizes network resources, but does not contribute services could have a huge impact on the efficacy of blue streaming systems, leading to scalability issues and service degradation. We discuss why BitTorrent-like tit-for-tat mechanisms cannot be simply tailored and used in streaming. Even though the problem of free riding is more severe in P2P media streaming than in file sharing, the deployed systems still lack incentive schemes. In this paper, we categorize, analyze, and compare a range of incentive mechanisms proposed for P2P streaming systems in the literature and discuss future research issues for these schemes to be deployed in practice.

Index Terms

peer-to-peer networks (P2P), media streaming, free riding, incentive mechanisms.

I. Introduction

As media files are usually large, media streaming places big demand on the bandwidth resources of streaming servers. Using a P2P network reduces the load on servers by exploiting bandwidths from participating peers. Free riding, where a peer utilizes network services but doesn’t contribute resources, hurts the performance of any P2P network and makes the streaming system hard to scale. In this paper, we will discuss and analyze a set of incentive schemes that are designed to fight against free riding. For us to better understand the analysis of incentive schemes, we will first give a brief tour of media streaming types, why P2P network is adopted for media streaming, and P2P streaming topologies.

Media streaming is available in two forms – live streaming and on-demand streaming. In live streaming, a user views a live event broadcasted in real time, for instance, watching President Obamas swearing-in ceremony while it was occurring. On the other hand, in on-demand streaming, a user views archived content at any point in time, like watching a film or documentary.

Mostly, client-server protocols are being used to provide streaming solutions. YouTube is a good example of this model: our client browsers request and stream from a central YouTube server. However, this means the problems inherent with a client-server architecture persist. It is highly dependent on the server's availability and bandwidth capacity.

Let us contrast the above with peer-to-peer (P2P) networks, in which peers not only request media packets from the server, but also supply these media packets to other peers. Because of peers' contributions, the system enjoys larger streaming capacity, is more resilient to peers' failures or departures, is more scalable, and is more economical in terms of setting up the streaming infrastructure. Thus, compared to a client-server model, the P2P model is better suited to meet the demanding requirements of media streaming applications.

II. P2P Streaming Topologies

As the design of incentive mechanisms in P2P streaming systems is usually applicable to a particular type of system topology, in this section, we will discuss existing topologies in popular P2P streaming systems. These topologies are applicable to both live and on-demand streaming systems. As shown in Fig. 1, current P2P streaming systems fall into two categories: tree-based and mesh-based.

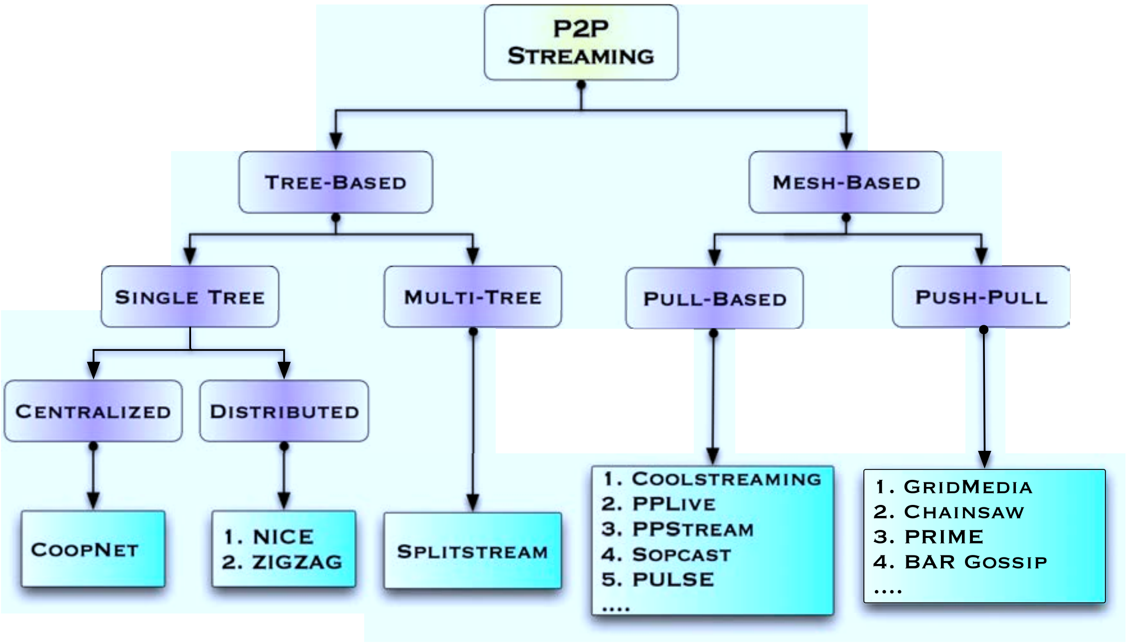

Fig. 1. A classification of current P2P streaming systems.

A **tree-based system**, is a hierarchical system, in which media packets originate from a root node, and are forwarded by internal peers to all the nodes in the tree. The leaf nodes, residing at the bottom of the tree, do not need to forward the packets any further.

There are two common criticisms of tree-based systems. First, the media streaming rate from the root to any node cannot exceed the minimum outgoing bandwidth of any internal node along the path. Therefore, an internal node with a small supplying bandwidth becomes the bottleneck. Second, the available bandwidths of the streaming system are not fully utilized, as the leaf nodes do not contribute any bandwidth.

One proposal to address the above problems is to build multiple streaming trees, as in SplitStream [1], where a node joins as a leaf in one tree and an internal node in the remaining trees.

A **mesh-based system** is an alternative P2P streaming architecture, which organizes peers in a dynamic mesh, where a peer receives media chunks from multiple nodes. Through the exchange of its buffer map, every peer node periodically advertises the availability of its media chunks. In a mesh-based system, the streaming paths are built entirely based on data requests. This eliminates the need to construct and maintain a fixed streaming topology, as in a tree-based system. CoolStreaming [2] is one of the first systems that advocates this data-driven design, and it quickly spurred the deployment of a number of other popular P2P systems, such as PPLive, PPStream, SopCast, PULSE, and many others.

In a mesh-based P2P system, peers are less vulnerable to network dynamics, as they pull media chunks from multiple peers, so the departures of a subset of peers have less detrimental effects on the streaming quality. The first batch of popular P2P systems adopted a pull-based design, where peers pull the desired media chunks by sending data requests. However, the periodic exchanges of buffer maps and the transmissions of data requests result in long latency in media playback. More recent P2P streaming systems, such as GridMedia [3], Chainsaw [4], PRIME [5], and BAR Gossip [6], adopt a hybrid pull-push based method, in which the system operates in alternating pull and push phases. In the pull phase, a peer receives media packets after sending request packets. Meanwhile, a packet transmission schedule is determined based on these request packets. In a push phase, packets are pushed out based on the same schedule found in the pull phase, without repeatedly exchanging buffer maps and transmitting data request packets. Among the systems, BAR Gossip is designed to tolerate selfish and malicious nodes.

TABLE I

COMPARISONS OF DIFFERENT STREAMING TOPOLOGIES.

	Single Tree	**Multi-Tree**	**Mesh**
resilience to peer churn	poor	medium	good
construction and maintenance	medium	large overhead	easy to maintain
bottleneck	low bandwidth internal nodes	low bandwidth internal nodes, but not as severe as in single tree	no
load balancing	no	among trees	among nodes
initial start-up delay	low	low	high

In Table I, we summarize the differences of the above streaming topologies. In short, a tree-based system is more efficient in pushing data to its participating peers in a short delay, while a mesh-based system is more resilient to peer churn and easier to scale to a larger size. This background of the underlying P2P network topologies would help us in understanding and analyzing the incentive mechanisms.

III. EVALUATION CRITERIA FOR INCENTIVE SCHEMES

Measurement studies on PPLive [7] and SopCast [7], [8], suggest that these deployed systems don't have incentive schemes to encourage contributions from all the peers. These studies clearly point to the need to develop incentive mechanisms to motivate a peer to contribute more resources. Before we discuss incentive mechanisms in depth, let us first identify the evaluation criteria for a good incentive scheme. Later, we will analyze each incentive mechanism using the criteria outlined below.

- **Tolerance to peer churn.** Peers may enter and leave the system at any time. If a supplying peer leaves in the middle of a transmission, the requesting peer must find a new supplier immediately. The incentive mechanism should consider the impact of peer churn in designing how to reward or penalize peers.

- **Accounting for heterogeneity in peers.** Peers participating in a streaming session are likely to be heterogeneous in many aspects: bandwidth capacity, playback time, computing resources, buffering capacity, and geographical location. It is challenging for an incentive mechanism to strike a delicate balance between the need and contributions of low-capacity and high-capacity peers and to accommodate all kinds of differences among peers.
- **Security against malicious peers.** Peers are self-interested and want to maximize their individual gains. Well-behaved peers will do so within the limits of built-in incentive schemes. Other malicious ones will like to exploit the incentive mechanisms, by misrepresenting their identities, defecting on their contributions, putting up false reputations, generating a large number of false identities, colluding with other peers, and by many other possible attacks yet to be discovered. Thus, the incentive mechanism should be trustworthy, robust against these attacks, and ensure fair exchanges in the system.
- **Topological considerations.** In both on-demand and live-streaming systems, peers are connected differently in single tree, multi-tree, and mesh-based topologies. An incentive mechanism needs to consider peers' upstream and downstream relationship to quantify peers' contribution levels and design an effective rewarding (or punishing) scheme. As a proposed incentive scheme doesn't need to work across all streaming topologies, we will clearly identify which streaming topology the scheme is designed for in our discussions.
- **Centralized or distributed.** The incentive mechanism should ideally have minimum dependencies. Thus, a distributed algorithm is more preferable over a centralized one, but simultaneously, the communication or computation overhead should not be too high.

IV. Incentive schemes

There are four types of incentive mechanisms specifically designed for P2P streaming system: reciprocal-based, reputation-based, game theory-based, and taxation-based schemes.

A. Reciprocal mechanisms

The reciprocal mechanisms follow the tit-for-tat strategy used by BitTorrent. Each peer measures the streaming rate of its neighbors, and sorts the peers per their upload rate. Periodically, choking decisions are made, in which the peer distributes its own limited uploading bandwidth accordingly, starting with the fastest uploader. Hence, a peer will upload more to the peers, which have contributed more to the peer. Thus, a free-rider that does not upload media packets to any peer suffers from poor streaming quality.

Reciprocal schemes have been used in both mesh and tree-based P2P streaming systems. The work in [9] adapts tit-for-tat idea to multi-tree based P2P streaming systems. In tree-based systems, a peer becomes a free rider when it stops forwarding packets to its downstream peers or when it refuses to accept any downstream peers. To address this problem, the authors propose to reconstruct multicast trees periodically. Such reconstructions are sufficiently different from the old ones so that the upstream and downstream relationship will change, thus it is impossible for a free rider to always act as a leaf node. PULSE [10], a mesh-based system, uses optimistic tit-for-tat to select peers to upload streaming packets. The policy helps find peers that have provided packets in recent past and that have expressed interest in the same playback window of streaming. Different from PULSE and other work that consider single layer videos, LayerP2P [11] streams layered videos in a mesh-based system. With layered videos, if a peer receives more video layers in the order of their importance, then it can videos of better quality. LayerP2P exploits this property and uses tit-for-tat strategy to send more video layers to a peer, who has supplied a large number of video layers in the recent past.

Let us analyze the reciprocal schemes using the evaluation criteria identified previously.

- **Tolerance to peer churn.** A peer depends on its private history of past measurements to decide whether to upload packets. Peer departures do not have big negative impact on the system, except for leaving some stale entries related to old peers. However, when a new peer joins the system, it raises a challenge of how to treat the new peer, as existing peers do not have past information on this new peer. A common idea is to grant it with a grace period, when the new peer will receive packets from the existing peers, without a record of past contributions. Hence, this opens door for exploitation by a selfish peer, so that it can become a free rider by frequently joining the system as a new peer.
- **Accommodation for heterogeneity in peers.** In streaming applications, especially on-demand streaming, peers have different playback times, and they exchange buffer maps to inform other peers about the availability of media packets. Consider a peer P_1 with a playback time T_1 and another peer P_2 with a playback time T_2, and $T_1 > T_2$. Obviously, P_1's buffer map contains the segments that are needed by P_2, while P_2's buffer map contains media segments that have already been viewed by P_1. If a reciprocal mechanism is strictly

enforced, P_1 will stream at a low rate to P_2, as P_2 is only able to serve P_1 with a limited set of media segments that are needed by P_1. As a result, using tit-for-tat, a peer is only able to get good comparable streaming service from fellow peers with similar playback times, similar contribution levels, and similar bandwidth capacity. In P2P streaming systems, when the demand on bandwidth contribution is very high, this reduces the effective set of supplying peers and results in worse streaming quality.

- **Security against malicious peers.** A reciprocal scheme is vulnerable when peers defect and choose to contribute less to the system. Consider the scenario in which a peer (say P_1) defects once and streams at a low rate to peer P_2. Consequently, peer P_2 will retaliate and send stream at a lower rate to P_1, which in turn streams at an even lower rate to peer P_2 in the future. This continues until collaboration dies between P_1 and P_2. This can result in collapse in system streaming capacity when the percentage of defected peers reaches a certain threshold. How to re-kindle collaboration is a big challenge.
- **Topological considerations.** Reciprocal schemes work for peers when media packets are sent and received in both directions. They work in both multi-tree and mesh-based systems. In single tree-based systems, media packets only flow in one direction, from upstream nodes to downstream nodes, and not vice versa. Based on reciprocal policy, upstream nodes would stop streaming to downstream nodes, because of the lack of their contributions. This totally breaks down the single-tree based streaming system.
- **Centralized or distributed.** Each peer maintains private history locally by calculating the streaming rates from each of its neighbor. Hence, the incentive mechanism operates locally at each peer, and are distributed.

B. Reputation-based Mechanisms

In a reputation-based system, a peer is assigned a score according to its contribution, and the score is subsequently mapped to a global rank (or reputation) in the system, determining the priority of the peer in receiving media service. This strategy differs from the reciprocal approach in that it relies on a global reputation to differentiate service and encourage cooperation.

In [12], when a peer with score S_i issues a request for a media segment, nodes with scores less than or equal to S_i will respond to the request. The score is decided by a peer's contribution, and then mapped into a percentile rank, based on the global distribution of scores. Therefore, a peer with a high percentile is likely to have a large set of candidate supplying peers to receive media segments from, and a free-rider, with a low percentile, would not be able to get packets from other peers. The PULSE [10] system also employs a history score to maintain all the past interactions with every other peer. This long-term history score is used to complement tit-for-tat in selecting peers.

Eigentrust [13] is another well known reputation-based algorithm. In Eigentrust, a peer ranks other peers and assigns each of them a local trust value. It calculates a peer's global reputation, similar to Google's PageRank algorithm, as a weighted average of the local trust values assigned by other peers. The weights are computed iteratively from the global reputation of the assigning peers. It may take many iterations to converge to global reputation values of peers. In P2P streaming, peer connections are highly dynamic and media segments are time-sensitive, so it is still an open question to decide whether the the algorithm's converging speed is fast enough to keep up with the connection dynamics and media playback.

Let us analyze reputations in more depth.

- **Tolerance to peer churn.** Similar to tit-for-tat schemes, it takes time for reputation to be established for new peers. An initial grace period is granted for new peers that can be exploited to free ride the system. This gives room for free riders to obtain unfair benefits.
- **Accommodation for heterogeneity in peers.** Like tit-for-tat, there is a service differentiation of peers. Peers with poor bandwidth resources are likely to have lower reputation scores in the system. In return, they will have lower priority in selecting which peers to obtain media packets from and suffer from poor streaming performance. On the other hand, peers with large bandwidth capacity have higher priority and enjoy better streaming performance.
- **Security against malicious peers.** Global reputation values are important in deciding whether to reward or punish peers, but the values are prone to manipulation by malicious peers. There are three common attacks: whitewashing attack, sybil attack, and collusion attack. In whitewashing attack, an attacker keeps changing identities to enjoy the benefits as a new comer, and thus escaping punishment. In a sybil attack, an attacker creates a large number of identities and uses these identities to receive a large share of system resources and gain large influence in reputation management. In collusion attack, a group of peers work together to misrepresent information and boost the reputation of every one in the collusion group. How to cope with these attacks is a big challenge for reputation schemes.

- **Topological considerations.** The reputation scheme can be applied in both tree-based and mesh-based systems. In a mesh-based system, a peer can simply use the reputation value to decide how to allocate its uploading bandwidth to its neighboring peers. For a non-leaf node in a tree-based system, Ye et al. [14] propose to periodically evaluate the contribution levels of its children peers. If one of its descendant peers defects and refuses to stream media segments down the tree, then the peer will select another peer, possibly a descendant of the defecting peer, as its child peer. Thus the topology of media streaming tree is dynamically changing based on peers' contribution levels.
- **Centralized or distributed.** In the systems that rely on a central service to calculate, maintain, and communicate the global reputation values, the central server is a potential bottleneck in streaming performance. In distributed schemes, converging to global trust values takes time.

C. Game Theoretic Mechanisms

In P2P streaming systems, peers are strategic players. They want to maximize their payoffs, *i.e.*, the quantity and quality of streaming packets received from the system, but meanwhile, reduce their cost, *i.e.*, the number of media packets to contribute to the system. Game theory is a popular modeling tool to study the strategic interactions among such rational players. Using the concepts and tools in game theory, one can derive when the strategy choices made by the peers become stable and no peer has an incentive to change from its equilibrium choice. This is very appealing in a dynamic P2P streaming environment.

The game usually consists of a set of N players, each player's available set of strategy choices, and each player's payoff function, as a result of the actions taken by the other players. In [15], Yeung and Kwok formulated a simple game to model packet exchange process between two players in the P2P streaming system. Here, each player needs to decide on the number of media packets to be sent to the other peer, and its payoff function is the streaming performance gain from receiving media packets from the other peer, subtracted by the cost of uploading packets. In every round, peers exchange packets among themselves and calculate their action for the next round. As their interactions continue as long as they stay in the system, this packet exchange process is modeled as an infinitely repeated game.

An important concept in game theory is Nash equilibrium. In the above game, it answers the follows questions: 1) when will the two players settle down on their choices for each round? and 2) what will be their equilibrium choice, given the knowledge of the other player's choice? In a simple two-player game, tit-for-tat has been shown to be an effective mechanism to model repeated interactions [16]. However, in P2P streaming systems, a peer interacts with a big set of neighboring peers. Theoretically, it is not clear whether tit-for-tat remains to be an effective mechanism in such a large distributed system. For larger systems, EquiCast [17] extends the game theory modeling to a system of N peers. It encourages cooperation using two schemes. The first scheme requires every peer p to keep a private history of every neighbor peer, q, as the balance between the number of media packets sent from q to p and the average link throughput. A node q is considered to be cooperative as long as its balance is greater than a predefined negative threshold, L. The second scheme imposes a penalty of one fine packet per round when the neighbor node q's balance goes below the threshold L. This forces peers to contribute to the level of expected link throughput in the system. The authors proved that when all the peers are selfish, every protocol-obeying strategy in which a peer cooperates with its neighbors is a Nash equilibrium.

Nash equilibrium is a concept in non-cooperative game theory. It does not model the scenario, when a set of peers form a coalition to seek better payoffs for the players in the coalition. Instead, cooperative game theory introduces solution concepts requiring that no set of players is able to break away and reap more payoffs for every player in the set. In P2P streaming, each peer chooses its upstream peers and downstream peers. The clustering of peers in the upstream/downstream relationship can be modeled as a coalition. In [18], Yeung and Kwok formulated a cooperative game to model how a parent selects its children peers and how a child selects the best parent based on its share of values.

Though theoretically elegant, game-theoretic schemes face a number of challenges.

- **Tolerance to peer churn.** In game-theoretic modeling, the derivations of Nash equilibrium strategy and the stable coalition have made an implicit assumption that the network topology is static and every peer stays in the system until the game ends. How each peer's game theoretic strategy performs in real-world scenario with frequent peer joins and departures is probably not tractable for theoretical analysis, but it can be evaluated through simulations and experiments. This should be an area of active research in the future.
- **Accommodation for heterogeneity in peers.** For the game theory models to be tractable,

peers are usually assumed to have identical payoff functions and strategy choices. This is clearly challenged in real systems, where peers differ in their bandwidth resources, storage capacities, and display sizes. Again, how heterogeneous peers perform using their game theoretic strategies remains to be evaluated through experiments.

- **Security against malicious peers.** The solution concepts in cooperative game theory help us understand when peers will benefit by forming a colluding group, rather than acting individually. It provides a useful theoretical tool to analyze whether an incentive scheme is subject to collusion attacks, but it is of limited use to analyze sybil and whitewashing attacks.
- **Topological considerations.** Game theoretic analysis has been used to model peer interactions in both tree-based and mesh-based topologies. In tree-based systems, the analysis provides guidelines for a parent peer to select its children such that the parent-child relationships are stable. In mesh-based systems, the analysis helps a peer to decide how many media packets to upload to each neighboring peer, based on a past history of their interactions.
- **Centralized or distributed.** To make an intelligent strategic choice, every peer needs to have complete information about every other peer in the system, including their payoff functions and set of possible strategy choices. The information can be obtained either from a central entity or through information exchanges among peers. Once every peer has sufficient information about the game, it makes its strategy choices independently, thus the execution of strategy by each peer is fully distributed.

D. Taxation Mechanisms

All the reciprocal, reputation, and game theory based algorithms differentiate peers based on their contributions. With such schemes in place, a peer with inherent less resources is only able to view a poor quality playback. Chu et.al [19] built a taxation scheme that relaxes this strict contribution-based differentiation. The taxation scheme motivates peers to contribute services that are commensurate with their resource levels, and requests peers with larger bandwidth to upload and contribute more to the system. By leveraging such altruistic behavior from resource rich peers, a taxation scheme improves the social welfare of the system, *i.e.*, the overall streaming quality perceived by all the users in the system.

Chu et.al built the taxation scheme in a streaming system using multi-tree topology. Every peer randomly selects a tree and joins the tree as an interior node, where it contributes to the system through sending the media packets to his descendants in the tree. It joins the remaining trees as leaf nodes, where it receives media packets from its upstream peers in these trees. So the peer assumes a supplier (or a consumer) role, when he works as an internal node (or a leaf node) in the tree. This simple relationship makes it easy to engineer taxation scheme in the system. To forward f unit bandwidth, a peer shall configure the fanout of the internal node to be f. To receive r unit bandwidth, the peer should join r trees. These calculations are done based on a tax schedule, which determines the upload rates of peers. A central authority, mostly the publisher of the media stream, exercises a tax for downloading, such that the net revenue of the system is optimal. The system has been successfully deployed to broadcast many events, including ACM SIGCOMM conferences, Carnegie Mellon commencement ceremonies, and distinguished lectures.

Taxation scheme is a relaxed form of incentive mechanism, and let us see how it works against our evaluation criteria.

- **Tolerance to peer churn.** As the authors pointed out, one possible objection to a taxation scheme is that mandatory taxation may take over and discourage voluntary contributions by altruistic peers. When altruistic peers leave, the remaining peers in the system will experience service degradation or even system collapse. In addition, like any other tree-based system, the departure of a peer leads to the reconstruction of streaming trees. This adds to the cost and streaming delays.
- **Accommodation for heterogeneity in peers.** The scheme aims to improve the overall streaming quality, perceived by heterogeneous peers, whether they are rich or poor in resources. This is certainly good news for resource poor peers, but it faces challenges in environments like enterprises and universities that host resource rich peers. Their peer nodes are usually leveraged as big contributors in the system. We have seen strong resistance to Skype, a popular P2P voice-over-IP application, from several universities. They ban Skype from campus network, partly because university nodes are often used as supernodes, forwarding a lot of traffic, in Skype network.
- **Security against malicious peers.** This scheme is applicable only when resource rich peers are willing to contribute more bandwidths to subsidize peers with slow Internet connections. This condition is true in a trusted environment, for example, within a university, where peers

are well aligned in their normal Internet activities. It may not work well in a more diverse environment with many malicious peers, attempting to exploit the system for more benefit.

- **Topological considerations.** The current implementation uses a multi-tree topology. For each peer, it easily maps the targeted level of the peer's bandwidth contributions to the number of downstream peers that the peer should have, and directly relates the peer's bandwidth gain to the number of trees that the peer joins as a leaf node. In a mesh-based system, the fanout degree of the peer and the number of upstream parents are largely driven by data requests. Each peer is bound to have different number of upstream and downstream peers. Hence, to track such information for every peer in a mesh-based system and to enforce a tax schedule will be a formidable task for a central server.
- **Centralized or distributed.** This scheme requires a central authority to impose the tax schedule. In case the publisher of the media stream assumes this role, the publisher must remain in the loop all the time, and is likely to be the bottleneck in the system.

V. Discussions and Summary

Table II presents a summary of P2P streaming incentive mechanisms. The reciprocal schemes are simple, intuitive, easy to implement, and widely deployed. In essence, each peer relies on its private history of other peers to decide how to interact with them. Relying on local history works well when a streaming object is popular, with many concurrent peers uploading and downloading the object.

In contrast, reputation-based schemes maintain a global history of a peer's past contribution to the entire P2P streaming network, rather than to individual peers. It provides a more persistent and precise measurement on past contribution levels. Global history can be built using either a centralized or a distributed method, like in EigenTrust. The biggest challenge for a reputation scheme is how to cope with malicious peers in sybil, whitewashing, and collusion attacks. EigenTrust, a potentially more robust scheme, is hard to deploy in real-time streaming systems, because the convergence to global reputation values takes too much time.

Both reciprocal and reputation schemes are simple and intuitive. Game theoretic approaches borrow the concepts of Nash equilibrium and stable coalition to model strategy choices chosen by the peers in the system. The approaches provide a beautiful theoretical foundation on when the system reaches an equilibrium state that no peer is willing to deviate from its equilibrium

TABLE II
SUMMARY OF P2P STREAMING INCENTIVE MECHANISMS.

	Incentive	**Key Advantages**	**Primary Concern**
Reciprocal	maintaining a private history of download rates and rewarding peers accordingly	simple, intuitive, easy to implement	data available only from similar peers
Reputation	using a global rank of peers to decide priority in selecting peers and desirable media segments	more accurate measurement of peers' contributions	vulnerable to a range of security threats
Game Theoretic	deriving peer strategy through game-theoretic modeling and analysis	Nash equilibrium and stable set concepts provide insights on when the system stabilizes and is resistant to collusion attacks	based on the assumptions that the network topology is static, every peer has the same payoff functions and strategy choice set
Taxation	peers with larger bandwidth making larger contributions to the system	encouraging a peer to contribute as much as it can, while improving the social welfare of the system	not eliminating free-riders and only applicable in trusted environments

choices. However, to derive such an equilibrium, one has to make a number of assumptions on static network connectivity, identical peer strategy set, and identical peer payoff functions.

Taxation schemes can be considered as relaxed incentive schemes. They require each peer to contribute as much as it can, commensurate with its resource levels. Consequently, resource-rich peers will shoulder more load in the streaming system, subsidizing those peers with slower uplink bandwidths. When used in trusted environments, where resource-rich peers are willing to contribute, taxation schemes improve the overall streaming performance of the system. The system does exercise a tax for downloading, however, it still leaves room for a free rider to exploit the contributions from altruistic peers.

As we can see, none of the incentive mechanisms is perfect and works in all scenarios. The P2P system designers should adopt the incentive schemes that meet their application requirements. We believe that the next generation P2P streaming systems will address the challenges in existing schemes. They should be fully distributed and scalable like reputation-based systems, should leverage sound theoretical tools to cope with security threats, and should be easy to implement and deploy in practice.

ACKNOWLEDGMENT

This work was supported in part by NSF CAREER (CNS-0546870) and NSF TRUST (CCF - 0424422) – Team for Research in Ubiquitous Secure Technology.

REFERENCES

[1] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, “Splitstream: high-bandwidth multicast in cooperative environments,” in *Proc. ACM Symposium on Operating Systems Principles (SOSP ’03)*, 2003, pp. 298–313.

[2] X. Zhang, J. Liu, B. Liz, and T.-S. P. Yum, “Coolstreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming,” in *Proc. IEEE INFOCOM*, Miami, Florida, Apr 2005.

[3] L. Zhao, J.-G. Luo, M. Zhang, W.-J. Fu, J. Luo, Y.-F. Zhang, and S.-Q. Yang, “Gridmedia: A practical peer-to-peer based live video streaming system,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, Nov 2005, pp. 1–4.

[4] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, A. E. Mohr, and E. E. Mohr, “Chainsaw: Eliminating trees from overlay multicast,” in *Proc. of the 4th International Workshop on Peer-to-Peer Systems (IPTPS’05)*, 2005, pp. 127–140.

[5] N. Magharei and R. Rajaie, “PRIME: peer-to-peer receiver-driven mesh-based streaming,” in *Proc. IEEE INFOCOM*, May 2007, pp. 1415–1423.

[6] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin, “BAR gossip,” in *Proc. the Seventh Symposium on Operating Systems Design and Implementation (OSDI)*, Seattle, WA, 2006.

[7] S. Ali, A. Mathur, and H. Zhang, “Measurement of commercial peer-to-peer live video streaming,” in *Proc. Int’l Workshop in Recent Advances in Peer-to-Peer Streaming*, Nov 2006.

[8] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, and S. Tewari, “Will IPTV ride the peer-to-peer stream?” *IEEE Communications Magazine*, vol. 45, pp. 86–92, Jun 2007.

[9] T. W. J. Ngan, D. S. Wallach, and P. Druschel, “Incentives-compatible peer-to-peer multicast,” in *Proc. Second Workshop on the Economics of Peer-to-Peer Systems*, July 2004.

[10] F. Pianese, D. Perino, J. Keller, and E. Biersack, “Pulse: An adaptive, incentive-based, unstructured p2p live streaming system,” *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1645 –1660, Dec 2007.

[11] Z. Liu, Y. Shen, K. Ross, S. Panwar, and Y. Wang, “LayerP2P: Using layered video chunks in p2p live streaming,” *IEEE Transactions on Multimedia*, vol. 11, no. 7, pp. 1340 –1352, Nov. 2009.

[12] A. Habib and J. Chuang, “Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming,” *IEEE Transactions on Multimedia*, vol. 8, no. 3, pp. 610–621, June 2006.

[13] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The EigenTrust algorithm for reputation management in P2P networks,” in *Proc. Int’l Conference on World Wide Web (WWW ’03)*, 2003, pp. 640–651.

[14] S. Ye and F. Makedon, “Collaboration-aware peer-to-peer media streaming,” in *Proc. ACM Int’l Conf. on Multimedia*, 2004, pp. 412–415.

[15] M. Yeung and Y.-K. Kwok, “Game-theoretic scalable peer-to-peer media streaming,” in *Proc. IEEE Int’l Symposium on Parallel and Distributed Processing (IPDPS 2008)*, April 2008, pp. 1–11.

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

[17] I. Keidar, R. Melamed, and A. Orda, “EquiCast: Scalable multicast with selfish users,” *Computer Networks*, vol. 53, no. 13, pp. 2373–2386, 2009.

[18] M. Yeung and Y.-K. Kwok, “Game theoretic peer selection for resilient peer-to-peer media streaming systems,” in *Proc. Int’l Conf. on Distributed Computing Systems (ICDCS ’08)*, June 2008, pp. 817–824.

[19] Y.-h. Chu, J. Chuang, and H. Zhang, “A case for taxation in peer-to-peer streaming broadcast,” in *Proc. ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, 2004, pp. 205–212.