Editor: Siobhán Clarke • siobhan.clarke@cs.tcd.ie

Incentive Mechanisms in P2P Media Streaming Systems

Xiao Su • *San Jose State University*
Suchreet K. Dhaliwal • *Cisco Systems*

Free riding, whereby a peer utilizes network resources but doesn't contribute services, could have a huge impact on the efficacy of peer-to-peer media streaming systems, leading to scalability issues and service degradation. Here, the authors categorize, analyze, and compare a range of incentive mechanisms proposed for P2P streaming systems.

Media files are usually quite large, and media streaming places a big demand on streaming servers' bandwidth resources. Using a peer-to-peer (P2P) network can reduce the load on servers by exploiting bandwidths from participating peers. *Free riding*, where a peer uses network services but doesn't contribute resources, hurts any P2P network's performance and makes the streaming system hard to scale. Various incentive schemes are designed to fight free riding, but to analyze these schemes, we must first understand media streaming types, why P2P networks are adopted for media streaming, and P2P streaming topologies.

Understanding Media Streaming

Media streaming is available in two forms — live and on-demand. In live streaming, a user views a live event broadcast in real time — for instance, watching US President Barack Obama's swearing-in ceremony while it was occurring. With on-demand streaming, a user views archived content at any point in time, as when watching a recorded film or television show.

Client–server protocols are used most to provide streaming solutions. YouTube is a good example of this model: client browsers request and stream from a central YouTube server. However, this means the problems inherent with a client–server architecture, such as server overload and bandwidth shortage, persist, and this approach depends substantially on the server's availability and bandwidth capacity.

In P2P networks, by contrast, peers can not only request media packets from the server but also from other peers. Because of peers' contributions, the system enjoys larger streaming capacity, is more resilient to peers' failures or departures, and is scalable and 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 media streaming applications' demanding requirements.

P2P Streaming Topologies

Incentive mechanisms for P2P streaming systems are typically designed with a particular system topology in mind. Several topologies exist in popular P2P streaming systems, and many apply to both live and on-demand streaming systems. As Figure 1 shows, current P2P streaming systems fall into two categories: tree-based and mesh-based.

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, don't need to forward the packets any further.

Tree-based systems experience two common criticisms. First, the media streaming rate from the root to any node can't exceed the minimum outgoing bandwidth of any internal node along the path. So, an internal node with a small supplying bandwidth becomes a bottleneck. Second, this approach doesn't fully utilize the streaming system's available bandwidth because leaf nodes don't contribute any bandwidth. One proposal

 1089-7801/10/$26.00 © 2010 IEEE

to address these 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.

Mesh-based systems organize peers in a dynamic mesh, in which a peer receives media chunks from multiple nodes. Using a buffer map, every peer node periodically advertises its media chunks' availability. 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 was one of the first systems to advocate this data-driven design,[2] and it quickly spurred the deployment of several other popular P2P systems, such as PPLive, PPStream, SopCast, and PULSE.

In a mesh-based P2P system, peers are less vulnerable to network dynamics. They pull media chunks from multiple peers, so a subset of peers departing has fewer detrimental effects on streaming quality. The first batch of popular P2P systems adopted a pull-based design, in which peers pull the desired media chunks by sending data requests. However, the periodic exchanges of buffer maps and 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 the pull phase, a peer receives media packets after sending request packets. Meanwhile, the other peers determine the packet transmission schedule based on these request packets. In the push phase, the peer pushes packets 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 the only one designed to tolerate selfish and malicious nodes.

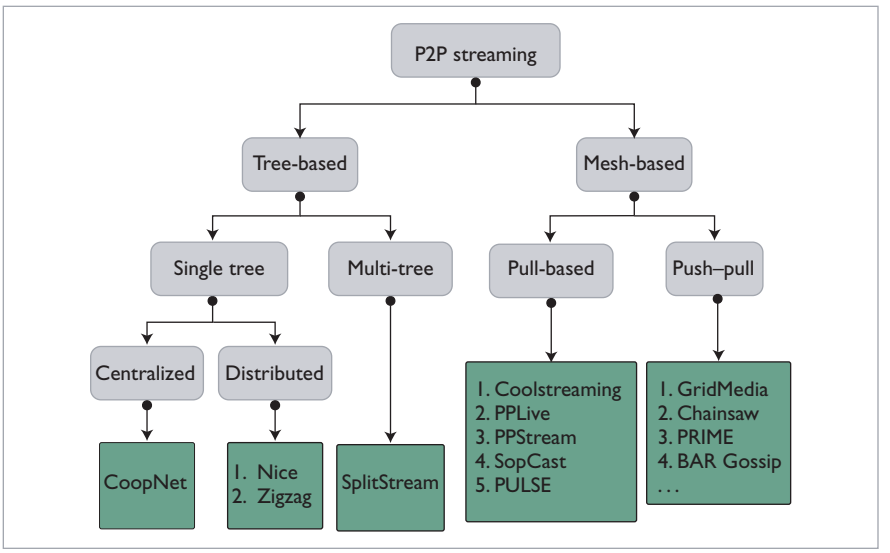

Figure 1. A classification of current peer-to-peer (P2P) streaming systems. P2P systems fall into two main categories: tree-based and mesh-based.

Table 1. Different streaming topologies.

	Single tree	Multi-tree	Mesh-based
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 the single tree	No
Load balancing	No	Among trees	Among nodes
Initial start-up delay	Low	Low	High

Table 1 summarizes the differences between the single tree, multi-tree, and mesh-based topologies. In short, a tree-based system is more efficient in pushing data to its participating peers with a short delay, whereas a mesh-based system is more resilient to peer churn and easier to scale to a larger size. This background can help us understand and analyze the incentive mechanisms.

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 peers. These studies clearly point to the need to develop such mechanisms to motivate peers to contribute more resources. Here, we identify several evaluation criteria for a good incentive scheme.

Tolerance to Peer Churn

Peers can enter and leave a system at any time. If a supplying peer leaves in the middle of a transmission, the requesting peer must find a new supplier immediately. Any 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. Striking a delicate balance between the needs and contributions of low- and high-capacity peers can be a challenge, as can accommodating the 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. Malicious ones will likely exploit incentive mechanisms by, for example, misrepresenting their identities, defecting on their contributions, putting up false reputations, generating numerous false identities, or colluding with other peers. 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 must consider peers' upstream and downstream relationships to quantify their contribution levels and design an effective rewarding (or punishing) scheme.

Centralized or Distributed Mechanism

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 shouldn't be too high.

Incentive Schemes

We can apply four specific incentive mechanisms to P2P streaming systems: reciprocal, reputation, game theory, and taxation.

Reciprocal Mechanisms

Reciprocal mechanisms follow the tit-for-tat strategy BitTorrent uses. Each peer measures its neighbors' streaming rate and sorts those neighbor peers based on their upload rate. Periodically, peers make unchoking decisions in which they distribute their own limited uploading bandwidth accordingly, starting with the fastest uploader. Hence, a peer will upload more to other peers that have contributed more to it. A free rider that doesn't upload media packets to any peer suffers from poor streaming quality.

Both mesh- and tree-based P2P streaming systems have used reciprocal mechanisms, and Tsuen-Wan Ngan and his colleagues adapt this tit-for-tat idea to multi-tree-based P2P streaming systems.[9] In tree-based systems, a peer becomes a free rider when it stops forwarding packets to its downstream peers or refuses to accept any downstream peers. To address this free-riding problem, the authors propose reconstructing multicast trees periodically. Such reconstructions are sufficiently different from the old trees to change the upstream and downstream relationship. Thus, a free rider can't always act as a leaf node.

PULSE, a mesh-based system, uses optimistic tit-for-tat to select peers to upload streaming packets.[10] The policy helps uncover peers that have provided packets in the recent past and expressed interest in the same streaming playback window. Unlike PULSE and other work that considers single-layer videos, LayerP2P streams layered videos in a mesh-based system.[11] With layered videos, if a peer receives more video layers in order of importance, then it can display videos of better quality. LayerP2P exploits this property and uses a tit-for-tat strategy to send more video layers to peers that have supplied a large number of video layers in the recent past.

We can use our evaluation criteria to assess reciprocal schemes.

Tolerance to peer churn. A peer depends on its private history of past measurements to decide whether to upload packets. In a reciprocal scheme, peer departures don't have a big negative impact on the system, except for leaving some stale entries related to old peers. However, when a new peer joins, it raises the question of how to treat that peer because existing peers don't have past information about it. A common idea is to grant the peer a grace period when it will receive packets from existing peers without a record of past contributions. This opens the door for a selfish peer to exploit the system, becoming a free rider by frequently joining as a new peer.

Accommodation for heterogeneity. As mentioned, in streaming applications, especially on-demand streaming, peers have different playback times, and they exchange buffer maps to inform other peers about media packet availability. Consider a peer P_1 with a playback time T_1 and another peer P_2 with a playback time T_2, where $T_1 > T_2$. P_1's buffer map contains the segments that P_2 needs, whereas P_2's buffer map contains media segments that P_1 has already viewed. If a reciprocal mechanism is strictly enforced, P_1 will stream at a low rate to P_2 because P_2 can only serve P_1 with a limited set of media segments that the latter needs. Consequently, using tit-for-tat, a peer can get good, comparable streaming service only from fellow peers with similar playback times, contribution levels, and bandwidth capacity. In P2P streaming systems, where the demand on bandwidth contribution is very

high, this limitation 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 — that is, choose to contribute less to the system. Consider a scenario in which a peer P_1 defects once and streams at a low rate to peer P_2. Peer P_2 will retaliate and send streams 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, which could result in a collapse in system streaming capacity when the percentage of defected peers reaches a certain threshold. Rekindling collaboration is a big challenge.

Topological considerations. Reciprocal schemes work for peers when peers send and receive media packets in both directions. They work in both multi-tree and mesh-based systems. In single tree-based systems, media packets flow only in one direction, from upstream nodes to downstream ones. Based on reciprocal policy, upstream nodes would stop streaming to downstream nodes due to a lack of contributions from the latter. This totally breaks down the single-tree-based streaming system.

Centralized or distributed mechanism. Each peer maintains a private history locally by calculating each of its neighbors' streaming rates. The incentive mechanism operates locally at each peer and is hence distributed.

Reputation-Based Mechanisms

Reputation-based systems assign a peer a score according to its contribution, subsequently mapping the score to a global rank (or reputation), which determines the peer's priority 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 one reputation-based system,[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. A peer's contribution determines the score, which the system then maps into a percentile rank based on the global distribution of scores. So, a peer with a high percentile is likely to have a large set of candidate supplying peers from which it can receive media segments, whereas a free rider with a low percentile wouldn't be able to get packets from other peers. The PULSE system also employs a history score to maintain all past interactions with every other peer.[10] PULSE uses this score to complement tit-for-tat in selecting peers.

EigenTrust is another well-known reputation-based algorithm,[13] in which a peer ranks other peers and assigns each one a local trust value. The algorithm calculates a peer's global reputation, similar to Google's PageRank algorithm, as a weighted average of the local trust values other peers have assigned it. EigenTrust then computes weights iteratively from the assigning peers' global reputation. Many iterations might be required to converge to peers' global reputation values. In P2P streaming, peer connections are highly dynamic, and media segments are time-sensitive, so deciding whether the algorithm's converging speed is fast enough to keep up with the connection dynamics and media playback is still an open question for P2P streaming system designers.

Let's analyze reputation-based systems in more depth.

Tolerance to peer churn. Similar to tit-for-tat schemes, it takes time for new peers to establish reputation. They receive an initial grace period that free riders can exploit to obtain unfair benefits from the system.

Accommodation for heterogeneity. Like tit-for-tat, a service differentiation exists among peers. Those with poor bandwidth resources are likely to have lower reputation scores in the system. In return, they'll have lower priority in selecting which peers to obtain media packets from and will 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 malicious peers can manipulate these values. Three common attacks occur: whitewashing, sybil, and collusion. In whitewashing, an attacker keeps changing identities to enjoy the benefits as a newcomer, thus escaping punishment. In a sybil attack, the attacker creates numerous identities and uses them to receive a large share of system resources and gain substantial influence in reputation management. In a collusion attack, a group of peers work together to misrepresent information and boost each group member's reputation. Coping with these attacks is a big challenge for reputation schemes.

Topological considerations. The reputation scheme is applicable to both tree- 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, Song Ye and colleagues propose periodically evaluating its children peers' contribution levels.[14] If a descendant peer defects and refuses to stream media segments down the tree, then the parent peer will select another peer, possibly a descendant of the defecting one, as its child. Thus,

the media streaming tree's topology changes dynamically based on peers' contribution levels.

Centralized or distributed mechanism. In systems that rely on a central service to calculate, maintain, and communicate global reputation values, the central server is a potential bottleneck in streaming performance. In distributed schemes, converging to global trust values takes time.

Game-Theoretic Mechanisms

In P2P streaming systems, peers are strategic players. They want to maximize their payoffs — that is, the quantity and quality of streaming packets received from the system — but simultaneously reduce their costs — the number of media packets they contribute to the system. Game theory is a popular modeling tool for studying strategic interactions among such rational players. Using its concepts and tools, we can derive when peers' strategy choices 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 other players take. Mark Kai Ho Yeung and Yu-Kwong Kwok formulated a simple game to model the packet-exchange process between

stay on the Cutting Edge of Artificial Intelligence

IEEE Intelligent Systems provides peer-reviewed, cutting-edge articles on the theory and applications of systems that perceive, reason, learn, and act intelligently.

The #1 AI Magazine
www.computer.org/intelligent
IEEE Intelligent Systems

two players in a P2P streaming system.[15] Here, each player must decide how many media packets to send to the other peer, and its payoff function is the streaming performance gain from receiving media packets back minus the cost of uploading packets. In every round, peers exchange packets among themselves and calculate their action for the next round. Because 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 game just described, this concept answers the following questions: When will the two players settle down on their choices for each round? What will be their equilibrium choice, given the knowledge of the other player's choice? In a simple two-player game, tit-for-tat is an effective mechanism.[16] However, in P2P streaming systems, a peer interacts with a big set of neighboring peers. It isn't clear whether tit-for-tat remains an effective mechanism in such a large distributed system. For larger systems, EquiCast extends game theory modeling to a system of N peers.[17] It encourages cooperation using two schemes. The first requires each peer, p, to keep a private history of every neighbor peer, q, as the difference between the number of media packets sent from q to p and the average link throughput. We can consider a node q to be cooperative as long as its balance is greater than a predefined negative threshold, L. The second scheme imposes a penalty of one 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 noncooperative game theory — it doesn't model a scenario in which a set of peers form a coalition to seek better payoffs for players. Cooperative game theory, on the other hand, introduces solution concepts requiring that no set of players be able to break away and reap more payoffs for every player in the set. In P2P streaming, each peer chooses its upstream and downstream peers. We can model this clustering of peers in an upstream/downstream relationship as a coalition. Yeung and Kwok also 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.[18]

Although theoretically elegant, game-theoretic schemes face several challenges.

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

Accommodation for heterogeneity. For game theory models to be tractable, we must usually assume that peers have identical payoff functions and strategy choices. Clearly, real systems challenge this assumption because peers differ in their bandwidth resources, storage capacities, and display sizes. Again, how heterogeneous peers perform using 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's of limited use in analyzing sybil and whitewashing attacks.

Topological considerations. Researchers have used game-theoretic analysis to model peer interactions in both tree- 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 decide how many media packets to upload to each neighboring peer based on a past history of their interactions.

Centralized or distributed mechanism. To make an intelligent, strategic choice, every peer must have complete information about every other peer in the system, including their payoff functions and possible strategy choices. Peers can obtain this information 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 each peer's strategy execution is fully distributed.

Taxation Mechanisms

Reciprocal, reputation, and game-theory-based algorithms differentiate peers based on their contributions. With such schemes in place, a peer with inherently fewer resources can view only a poor-quality playback. Yang-Hua Chu and his colleagues built a taxation scheme that relaxes this strict contribution-based differentiation.[19] The scheme motivates peers to

Table 2. 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	Provides insights on when the system stabilizes and is resistant to collusion attacks	Assumes that the network topology is static and that every peer has the same payoff functions and strategy choice set
Taxation	Having peers with larger bandwidth contribute more to the system	Encourages peers to contribute as much as they can while improving the system's social welfare	Doesn't eliminate free riders and is only applicable in trusted environments

contribute services that are commensurate with their resource levels and requests that those with larger bandwidth upload and contribute more to the system. By leveraging such altruistic behavior from resource-rich peers, a taxation scheme improves the system's social welfare — that is, the overall streaming quality that all users in the system perceive.

The authors built the taxation scheme in a streaming system using a multi-tree topology. Every peer randomly selects a tree and joins it as an interior node, where it contributes to the system by sending media packets to its descendants in the tree. The peer joins the remaining trees as a leaf node, where it receives media packets from its upstream peers. So, the peer assumes a supplier role when working as an internal node and a consumer role when acting as a leaf node. This simple relationship makes it easy to engineer a taxation scheme in the system. To forward f unit bandwidth, a peer shall configure the internal node's fanout (that is, the number of outgoing branches) to be f. To receive r unit bandwidth, the peer should join r trees. We make these calculations based on a tax schedule, which determines peers' upload rates. A central authority, usually the media stream's publisher, exercises a tax for downloading, such that the system's net revenue is optimal. The authors have successfully deployed this system to broadcast many events, including ACM SIGCOMM conferences, Carnegie Mellon commencement ceremonies, and distinguished lectures.

Taxation schemes are a relaxed form of incentive mechanism, but let's see how they work against our evaluation criteria.

Tolerance to peer churn. As Chu and colleagues pointed out,[19] one possible objection to a taxation scheme is that mandatory taxation might take over and discourage voluntary contributions from 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, a peer's departure leads to streaming tree reconstruction. This increases costs and streaming delays.

Accommodation for heterogeneity. Taxation schemes aim to improve overall streaming quality for heterogeneous peers, whether they're rich or poor in resources. This is certainly good news for resource-poor peers, but it presents challenges in environments such as enterprises and universities that host resource-rich ones. Their peer nodes are usually leveraged as big contributors in the system. We've seen strong resistance to Skype, a popular P2P voice-over-IP application, from several universities, which ban the application from campus networks partly because the Skype network often uses university nodes as supernodes, forwarding substantial traffic through the university network.

Security against malicious peers. We can apply taxation schemes only when resource-rich peers are willing to contribute more bandwidth to subsidize those 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 might not work well in a more diverse environment with many malicious peers attempting to exploit the system for more benefit, however.

Topological considerations. The current taxation 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 how many trees that peer joins as a leaf node. In a mesh-based system, the peer's fanout degree and number of upstream parents are largely driven by data requests.

Each peer is bound to have a different number of upstream and downstream peers. Hence, tracking such information for every peer in a mesh-based system and enforcing a tax schedule would be a formidable task for a central server.

Centralized or distributed mechanism. This scheme requires a central authority to impose the tax schedule. If a media stream publisher assumes this role, it must remain in the loop at all times and is likely to be a bottleneck in the system.

As we can see, no single incentive mechanism is perfect and works in all scenarios. Table 2 presents a summary of the P2P streaming incentive mechanisms, including their advantages and concerns. P2P system designers should adopt incentive schemes that meet their application requirements. The next generation of P2P streaming systems should address the challenges in existing schemes — they should be fully distributed and scalable, should leverage sound theoretical tools to cope with security threats, and should be easy to implement and deploy in practice.

Acknowledgments

This work was supported in part by US National Science Foundation Faculty Early Career Development (CAREER) grant CNS-0546870 and Team for Research in Ubiquitous Secure Technology (TRUST) grant CCF-0424422.

References

1. M. Castro et al., "Splitstream: High-Bandwidth Multicast in Cooperative Environments," *Proc. ACM Symp. Operating Systems Principles* (SOSP 03), ACM Press, 2003, pp. 298–313.
2. X. Zhang et al., "CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming," *Proc. IEEE Int'l Conf. Computer Communications* (INFOCOM 05), IEEE Press, 2005, pp. 2102–2111.
3. L. Zhao et al., "GridMedia: A Practical Peer-to-Peer-Based Live Video Streaming System," *Proc. IEEE Workshop on Multimedia Signal Processing*, IEEE Press, 2005, pp. 1–4.
4. V. Pai et al., "Chainsaw: Eliminating Trees from Overlay Multicast," *Proc. 4th Int'l Workshop Peer-to-Peer Systems* (IPTPS 05), LNCS, Springer, 2005, pp. 127–140.
5. N. Magharei and R. Rajaie, "PRIME: Peer-to-Peer Receiver-Driven Mesh-Based Streaming," *Proc. IEEE Int'l Conf. Computer Communications* (INFOCOM 07), IEEE Press, 2007, pp. 1415–1423.
6. H.C. Li et al., "BAR Gossip," *Proc. 7th Symp. Operating Systems Design and Implementation* (OSDI), Usenix Assoc., 2006, pp. 191–204.
7. S. Ali, A. Mathur, and H. Zhang, "Measurement of Commercial Peer-to-Peer Live Video Streaming," *Proc. Int'l Workshop Recent Advances in Peer-to-Peer Streaming*, 2006.
8. A. Sentinelli et al., "Will IPTV Ride the Peer-to-Peer Stream?" *IEEE Comm.*, vol. 45, June 2007, pp. 86–92.
9. T.W.J. Ngan, D.S. Wallach, and P. Druschel, "Incentives-Compatible Peer-to-Peer Multicast," *Proc. 2nd Workshop on the Economics of Peer-to-Peer Systems*, 2004, pp. 2–7.
10. F. Pianese et al., "Pulse: An Adaptive, Incentive-Based, Unstructured P2P Live Streaming System," *IEEE Trans. Multimedia*, vol. 9, no. 8, 2007, pp. 1645–1660.
11. Z. Liu et al., "LayerP2P: Using Layered Video Chunks in P2P Live Streaming," *IEEE Trans. Multimedia*, vol. 11, no. 7, 2009, pp. 1340–1352.
12. A. Habib and J. Chuang, "Service Differentiated Peer Selection: An Incentive Mechanism for Peer-to-Peer Media Streaming," *IEEE Trans. Multimedia*, vol. 8, no. 3, 2006, pp. 610–621.
13. S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," *Proc. Int'l Conf. World Wide Web* (WWW 03), ACM Press, 2003, pp. 640–651.
14. S. Ye and F. Makedon, "Collaboration-Aware Peer-to-Peer Media Streaming," *Proc. ACM Int'l Conf. Multimedia*, ACM Press, 2004, pp. 412–415.
15. M. Yeung and Y.-K. Kwok, "Game-Theoretic Scalable Peer-to-Peer Media Streaming," *Proc. IEEE Int'l Symp. Parallel and Distributed Processing* (IPDPS 08), IEEE Press, 2008, pp. 1–11.
16. M.J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.
17. I. Keidar, R. Melamed, and A. Orda, "EquiCast: Scalable Multicast with Selfish Users," *Computer Networks*, vol. 53, no. 13, 2009, pp. 2373–2386.
18. M. Yeung and Y.-K. Kwok, "Game-Theoretic Peer Selection for Resilient Peer-to-Peer Media Streaming Systems," *Proc. Int'l Conf. Distributed Computing Systems* (ICDCS 08), IEEE Press, 2008, pp. 817–824.
19. Y.-H. Chu, J. Chuang, and H. Zhang, "A Case for Taxation in Peer-to-Peer Streaming Broadcast," *Proc. ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, ACM Press, 2004, pp. 205–212.

Xiao Su is an associate professor in the computer engineering department at San Jose State University. Her research interests include network security, multimedia communications, media coding, and mobile computing. Su has a PhD in computer science from the University of Illinois at Urbana-Champaign. She's a recipient of the US National Science Foundation CAREER award and a senior member of IEEE and the ACM. Contact her at xiao.su@sjsu.edu.

Suchreet K. Dhaliwal is a software engineer in Cisco Systems' Metro Ethernet Team of Network Software and Systems Technology, working on Cisco's next-generation operating system, NXOS. She has an MS in software engineering from San Jose State University. Dhaliwal received a Silver Medal award for outstanding performance from the Punjab Engineering College (PEC), Chandigarh, India. Contact her at suchreetkaur@gmail.com.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.