

# Bandwidth and Latency Aware Contribution Estimation in P2P Streaming System

Cheng-Yun Ho, Ming-Hsiang Huang, Cheng-Yuan Ho, and Chien-Chao Tseng, *Member, IEEE*

**Abstract**—In this article, a bandwidth and latency aware contribution estimation (BLACE) mechanism is proposed to enhance the efficiency of content distribution in a mesh peer-to-peer (P2P) streaming system. This is because the mesh P2P streaming system outperforms others in many aspects, but most content distribution mechanisms use only one of upload bandwidth and link latency as a factor, several contribution-aware schemes (CASs) just punish selfish peers, and some CASs only rearrange the content receiving order. Hence, the BLACE proposes a prevention way considering every peer's upload bandwidth, link latency, and the maximum playing delay of a stream at the same time to prioritize the content distribution. Moreover, BLACE is implemented on Coolstreaming with NS2 and experimental results show that BLACE could at the most shorten the receiving time about 15.44%, 13.94%, and 8.26%, and improve the whole performance about 7.05%, 2.81%, and 2.69% at the most, when it compares with the Coolstreaming, the Coolstreaming with bandwidth preference, and the Coolstreaming with latency preference, respectively.

**Index Terms**—Bandwidth, latency, contribution estimation, BLACE, P2P streaming system.

## I. INTRODUCTION

PEER-TO-PEER (P2P) networks have grown aggressively in Internet traffic and increased greatly in popularity over the past decade. Familiar applications used in P2P networks are as file sharing and real-time multimedia streaming. In addition, a P2P real-time multimedia streaming application needs to solve the timing issue because the users will feel uncomfortable when the contents do not arrive in/on time. Accordingly, several P2P streaming systems have been designed [1]–[8] and some of them [2]–[8] focus on the system structure and content distribution.

Existing P2P streaming systems can be classified into three models: single-tree, multi-tree, and mesh [9]. In single-tree, the peer having the original stream is both an initiator and the root of the tree [1]. On the other hand, other peers form remaining parts of the tree by some defined formation method and receive the contents of stream in order. Similar to the single-tree, there is only one initiator in the multi-tree. However, the initiator divides the stream into multiple sub-streams, other peers construct the corresponding multiple sub-trees, and one sub-stream is transmitted through one sub-tree. Different from both single-tree and multi-tree, in mesh, the stream is divided into a lot of blocks with a defined size (called chunks), and each

peer randomly connects to others and exchanges which chunks it owns with its connected peers (called neighbors) [4], [5], [8].

Every model has its nature advantages and drawbacks, but R. Kumar *et al.* [10] proved that in theory, there is an upper bound for the performance of each P2P streaming model. N. Magharei *et al.* [11] further showed that the mesh has a better performance than trees. However, the mesh still has many issues, such as the limited availability of future content and in-time requirement of content delivery and hence, several mechanisms are proposed to solve these problems. For example, S. Xie *et al.* proposed a sub-stream based mesh P2P streaming system and called it Coolstreaming [2] to solve in-time issue and reduce the content delivery delay and the number of exchanging messages.

Most proposed mechanisms used either upload bandwidth or link latency as a serious factor to solve the mentioned problems. Furthermore, some researchers have similar ideas. After peers distribute contents, the mechanism periodically evaluates every peer's contribution. Based on the contribution, the mechanism either punishes selfish peers by degrading their orders [4], [8] or promotes peers with larger contribution closer to the source peer [5]–[8]. Unfortunately, even their experiment results presented that the proposed mechanisms have better outcomes and work well; it still has room to further improve the whole P2P performance and efficiency. Hence, in this article, to speed up the content distribution, a bandwidth and latency aware contribution estimation (BLACE) mechanism is proposed to use a prevention way that considers every peer's upload bandwidth, link latency, and the maximum playing delay of a stream at the same time to evaluate whether this peer could be a possible contributor before the peer transmits contents to other peers.

The rest of this article is organized as follows. The proposed mechanism, BLACE, is described in Section II. Section III shows experiment results and findings. Finally, the last section concludes this article and points out the future works.

## II. BANDWIDTH AND LATENCY AWARE CONTRIBUTION ESTIMATION MECHANISM

In this section, the BLACE is described in the first half while its model and equations are given in the second half.

### A. Detailed Descriptions of BLACE

In BLACE, it is assumed that every peer exchanges its upload bandwidth and link latency periodically with its neighbors (i.e., connected peers). In practice, this assumption is similar to the information sharing between nodes in reputation systems [12], [13] and can be realized by using the methods in [3], [12], [13].

The concept of the BLACE is that before a peer including the initiator distributes a chunk to other peers, it needs to calculate how many receivers it can support based on its available upload bandwidth, estimate every receiver's contribution, and selects the receivers who have a larger contribution. The contribution of a peer  $P$  for a chunk means that the number of peers which

Manuscript received March 9, 2014; accepted July 13, 2014. Date of publication July 28, 2014; date of current version September 8, 2014. The associate editor coordinating the review of this paper and approving it for publication was P. D. Yoo. (*Corresponding author: Chien-Chao Tseng.*)

C.-Y. Ho, M.-H. Huang, and C.-C. Tseng are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: cyho@cs.nctu.edu.tw; mhhuang@cs.nctu.edu.tw; cctseng@cs.nctu.edu.tw).

C.-Y. Ho is with the Advanced Research Institute, Institute for Information Industry (III), Taipei 105, Taiwan (e-mail: tommyho@iii.org.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LCOMM.2014.2343612

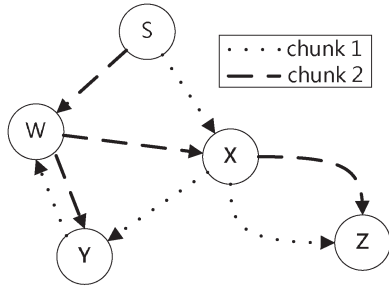


Fig. 1. An example of the contribution of peer X.

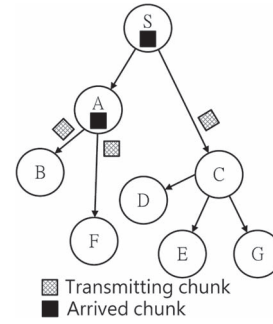


Fig. 3. Transmitting and arrived chunks.

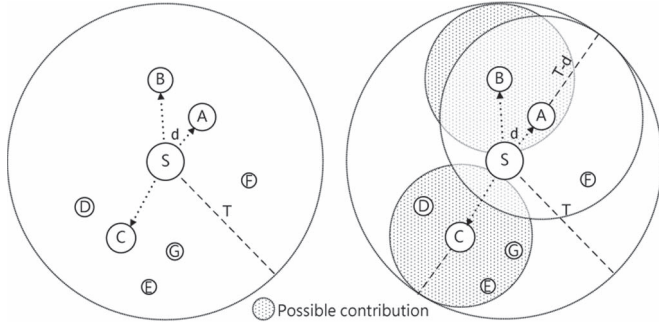


Fig. 2. A case of contribution estimation by peer S.

receive that chunk directly from the peer  $P$ . In Fig. 1, for example, the contributions of peer  $X$  for chunks 1 and 2 are 2 and 1, respectively, where the  $S$  denotes the source peer and the  $W, X, Y,$  and  $Z$  represent other peers. In addition, Fig. 2 demonstrates a case of contribution estimation by the peer  $S$ . The detailed descriptions are as follows.

In the left side of Fig. 2, the initiator is the peer  $S$ , other peers are receivers of the peer  $S$ , the circle size of a peer is the upload bandwidth size of that peer (i.e., the peer with a large circle has larger upload bandwidth than the peer with a small circle does), the length of dotted line with arrow means the link latency from a peer to another peer (e.g., the link latency from the peer  $S$  to the peer  $A$  is  $d$ ), and the threshold of the maximum playing delay is  $T$  that is presented by the circle with dotted line. Furthermore, the circle size and the length of dotted line with arrow will be varied by the condition of the network and time.

Before the peer  $S$  sends a chunk to receivers, it first divides its available upload bandwidth by the chunk size and gets 2 (i.e., the peer  $S$  only can support 2 peers presently). Next, as shown in the right side of Fig. 2, the peer  $S$  estimates every neighbor's contribution. The shadow of a circle is the possible contribution of a peer because it means that after a peer receives the chunk from the peer  $S$ , with enough upload bandwidth and before the delay exceeds the threshold  $T$ , how many receivers the peer can distribute that chunk to. In contrast to the possible contribution, the contribution is with the real available upload bandwidth. Moreover, the value of contribution is not greater than that of the possible contribution. For example, if the peer  $S$  transmits a chunk to the peer  $A$ , it will spend  $d$  time and the remained time for the peer  $A$  to distribute the chunk to other peers will be  $T-d$  time. Therefore, the peer  $A$  can send the chunk to both the peers  $B$  and  $F$  if it has enough upload bandwidth. In this case, the possible contributions of the peers  $A, B,$  and  $C$  are 2, 1, and 3, respectively. On the other hand, if the peer  $A$ 's available upload bandwidth is only 1 chunk, it will send the chunk to either the peer  $B$  or the peer  $F$ . Suppose that, in this case, the contribution equals to the possible contribution. Finally, the peer  $S$  selects the peers having a larger contribution to be its

TABLE I  
NOTATIONS AND DEFINITIONS

Notation	Definition	Notation	Definition
$N$	Number of peers in P2P system	$L(P)$	Link latency of peer $P$ in uploading direction
$B(P)$	Upload bandwidth of peer $P$	$\bar{L}$	Average link latency of all peers in uploading direction
$\bar{B}$	Average upload bandwidth of all peers	$\lambda$	Reciprocal of average link latency, $\lambda = 1/\bar{L}$
$T_c(t)$	Number of transmitting chunks for a chunk $c$ sent from source peer after $t$ seconds	$\rho(P)$	Number of receivers supported by peer $P$ (i.e., number of transmitting chunks created by peer $P$ )
$T(t)$	Normalized $T_c(t)$ for sub-stream or stream	$\bar{\rho}$	Average number of created transmitting chunks
$A_c(t)$	Number of arrived chunks for a chunk $c$ sent from source peer after $t$ seconds	$S_N$	Number of sub-streams
$A(t)$	Normalized $A_c(t)$ for sub-stream or stream	$S_R$	Bit rate of sub-stream

receivers. Hence, in this case, the peer  $S$  will choose peers  $C$  and  $A$ .

Next, the peers  $C$  and  $A$  execute the same processes as the peer  $S$  did before to select their own receivers and the selected ones will repeat the processes until all peers have been selected or the delay exceeds the threshold  $T$ . Consequently, the BLACE could be thought one kind of recursive mechanism. In the case of Fig. 2, suppose that the peer  $A$  chooses peers  $B$  and  $F$  while the peer  $C$  selects peers  $D, E,$  and  $G$  to be their receivers. All peers in Fig. 2 have been selected, so BLACE has done its work for this chunk and the final transmission path is shown as Fig. 3.

B. Model and Equations of BLACE

Based on the descriptions in the above sub-section, the contribution of a peer with the BLACE will be proved by the following model and equations. To clearly explain the modeling process, the final transmission path in the prior case is used as the reference. As shown in Fig. 3, the chunk on the transmission path is called the transmitting chunk while the chunk received by a peer is named the arrived chunk. Moreover, when all peers receive the chunk, the number of the transmitting chunk and the arrived chunk will be 0 and the number of peers, respectively.

Table I exhibits the notations and their definitions used in the following paragraphs.  $T_c(t)$  and  $A_c(t)$  are the number of the transmitting chunks and arrived chunks for a chunk  $c$  sent from source peer after  $t$  seconds. In fact,  $A_c(t)$  also means the number of peers which have received the chunk  $c$  after  $t$  seconds. Besides,  $T_c(t)$  and  $A_c(t)$  respectively equal to 0 and 1 when  $t$  is 0 (i.e.,  $T_c(0)=0$  and  $A_c(0)=1$ ). The average link latency  $\bar{L}$  is counted from every peer's link latency  $L(P)$  and the number of peers  $N$  (i.e.,  $\bar{L} = \sum L(P)/N$ ). Therefore, a transmitting chunk needs  $\bar{L}$  seconds to be an arrived chunk averagely. In addition, if a chunk is transmitted after  $\Delta t$  seconds, it will be an arrived

chunk with the probability  $\lambda\Delta t$ , where  $\lambda$  equals to  $1/\bar{L}$  and the probability is 100% when  $\Delta t$  is larger than  $\bar{L}$ . Accordingly, after  $\Delta t$  seconds, the equation for calculating the number of arrived chunks in the mesh P2P streaming system is denoted as

$$\begin{cases} A_c(t) = 1, & \text{if } t = 0 \\ A_c(t + \Delta t) = A_c(t) + T_c(t)\lambda\Delta t, & \text{otherwise.} \end{cases} \quad (1)$$

Similarly, the number of transmitting chunks is changed after  $\Delta t$  seconds. The equation is denoted as

$$\begin{cases} T_c(t) = 0, & \text{if } t = 0 \text{ or} \\ T_c(t) = B(S)/(S_N * S_R), & A_c(t) = N \\ T_c(t + \Delta t) = T_c(t)(1 - \lambda\Delta t) + \bar{\rho}T_c(t)\lambda\Delta t, & \text{otherwise} \end{cases} \quad (2)$$

where  $\bar{\rho}$  is the average number of created transmitting chunks. Moreover,  $\bar{\rho}$  is obtained from one of the coming equations:  $\bar{\rho} = [(\sum_{P=1}^N \rho(P))/N]$  or the normalized one,  $\bar{\rho} = [\bar{B}/(S_N * S_R)]$ , where  $\rho(P)$  is the number of receivers is supported by peer  $P$  (i.e., the number of transmitting chunks is created when peer  $P$  receives a chunk),  $\bar{B}$  is the average upload bandwidth of all peers,  $S_N$  is the number of sub-streams, and  $S_R$  is the bit rate of a sub-stream. Moreover,  $\rho(P)$  and  $\bar{B}$  equal to  $[B(P)/(S_N * S_R)]$  and  $\sum B(P)/N$ , respectively. Note that  $\bar{\rho}$  and  $\rho(P)$  are not smaller than 0.

The normalized  $T(t)$  and  $A(t)$  can be derived from Eqs. (1) and (2), and boundaries, such as  $T_c(0)$ , and  $A_c(0)$ , as follows.

$$T(t) = e^{\lambda\bar{\rho}t} \quad (3)$$

$$A(t) = \frac{1}{\bar{\rho}} e^{\lambda\bar{\rho}t} - \frac{1}{\bar{\rho}} \quad (4)$$

Suppose that a peer  $P$  receives a sub-stream or a stream at the  $t_p^{th}$  second. The amount of distributed chunks by the peer  $P$  from the time  $t_p$  to the time  $T$  can be derived from the number of peer  $P$ 's receivers and their received chunks (i.e.,  $\rho(P) * A(t)$ ) as the following expression (5).

$$\frac{B(P)}{S_N * S_R} A(T - t_p) \quad (5)$$

### III. PERFORMANCE EVALUATION

To observe how much performance the BLACE can improve in a P2P streaming system, we chose NS2 [14] as a simulation tool, implemented the BLACE into Coolstreaming, and executed a lot of experiments. The reasons for choosing NS2 and Coolstreaming are as follows. 1) S. Naicken *et al.* [15] investigated many existing P2P simulators, such as PlanetLab [16] and OverSim [17], and pointed out that only few simulators were designed for P2P streaming applications since most of them focused on providing a convenient user interface (UI) for either distributed hash table (DHT) or gossip-based searching algorithm; 2) NS2 has numerous networking modules including P2P streaming applications; 3) NS2 is a well-known simulator in the academic community; and 4) Coolstreaming is not only the first mesh P2P streaming system with sub-streams but also widely used in today's Internet.

#### A. Experiment Environments and Parameter Settings

Fig. 4 is a diagram of network topologies used in the following experiments. Every network topology is formed by 100 routers and 1001 peers (1 source peer and 1000 peers). The routers and the peers present the cores and the edges of the network, respectively. Furthermore, the 100 routers randomly

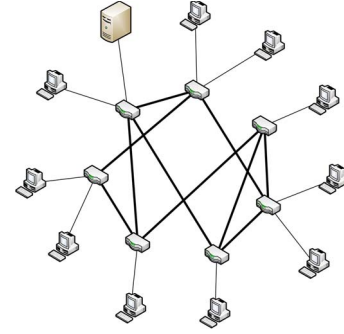


Fig. 4. A diagram of network topologies.

TABLE II  
PARAMETER SETTINGS

Parameter	Value	Parameter	Value
No. of sub-streams	8	Bit-rate of stream (sub-stream)	384 Kbps (48 Kbps)
Chunk size	4.8 Kbyte	Chunk length	0.1 seconds
No. of routers	100	No. of peers	1001
Bandwidth of core link	100-1000 Mbps	Bandwidth of edge link	300-900 Kbps
Latency of core link	0.001-0.1 seconds	Latency of edge link	0.001-0.1 seconds
Bandwidth between source peer and router	5 Mbps	Experiment duration	3600 seconds

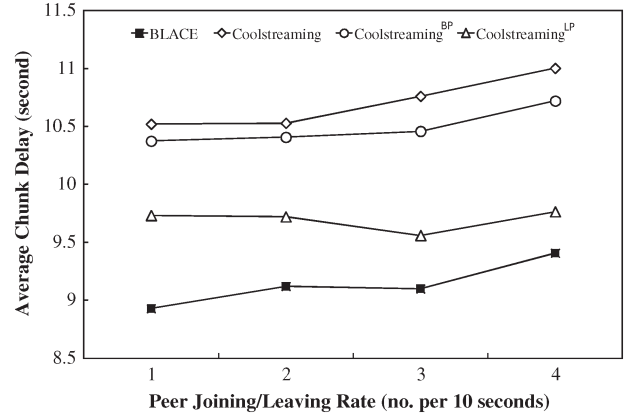


Fig. 5. Average chunk delay with different peer joining/leaving rates.

connect to each other and the links between routers are called core links. Similarly, each peer connects to a router randomly and the link between the peer and the router is called edge link. All links' latencies are from 0.001 seconds to 0.1 seconds and all core links' bandwidths are from 100 Mbps to 1000 Mbps. The bandwidth of edge link between source peer to router is 5 Mbps while other edge links' bandwidths are from 300 Kbps to 900 Kbps. This is because when the source peer supports more receivers, the BLACE can demonstrate its characteristics early (i.e., quickly distributes the stream to the entire P2P system). More parameter settings are illustrated in Table II.

#### B. Experiment Results and Findings

Both Figs. 5 and 6 exhibit the experiment results of four content distribution mechanisms executing on the above described networks with variable joining/leaving peers. The four mechanisms are BLACE, Coolstreaming, Coolstreaming with bandwidth preference (Coolstreaming<sup>BP</sup>), and Coolstreaming with latency preference (Coolstreaming<sup>LP</sup>), and for every experiment, the peer joining/leaving rate is fixed (e.g., 1, 2, 3, and 4 per 10 seconds). Moreover, the Coolstreaming<sup>BP</sup> and Coolstreaming<sup>LP</sup> denote the Coolstreaming respectively uses

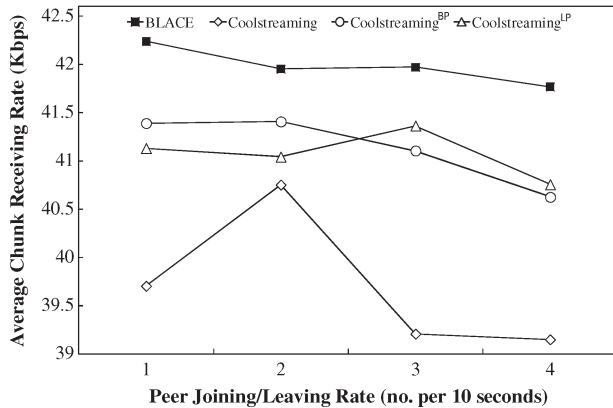


Fig. 6. Average chunk receiving rate with different peer joining/leaving rates.

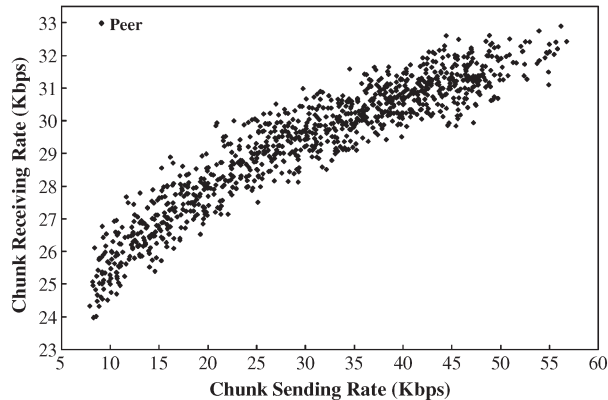


Fig. 7. Chunk receiving rate versus chunk sending rate.

upload bandwidth and link latency as the only factor to select receivers.

In Fig. 5, compared with the Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup>, the BLACE shortens the chunk delay (receiving time) about 15.14%, 13.94%, and 8.26% when there is 1 peer joining/leaving the network every 10 seconds. In addition, when more peers join/leave the network, the receiving time will be naturally increased no matter which mechanism is used. However, the BLACE still has the littlest chunk delay among these four mechanisms. For example, while the peer joining/leaving rate is 4 peers per 10 seconds, the chunk delay of the BLACE, Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup> is 9.4, 11.0, 10.7, and 9.8 seconds, respectively.

On the other point of view, as shown in Fig. 6, the BLACE has better performance for every sub-stream than other content distribution mechanisms. For example, compared with the Coolstreaming, the BLACE improves the average chunk receiving rate about 6.38%, 2.94%, 7.05%, and 6.69% when the peer joining/leaving rate is from 1 to 4 peers per 10 seconds, respectively. Even in the worst case (i.e., 2 peers join/leave the network every 10 seconds), for every sub-stream, the BLACE still achieves 1.03, 1.01, and 1.02 times higher average chunk receiving rate in comparison with the Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup>, respectively. Accordingly, the BLACE could make the whole P2P system more efficient.

In detail, Fig. 7 illustrates all peers' relationships between their chunk receiving rate and chunk sending rate in some sub-stream. From Fig. 7, the coordinates of the highest, lowest, most right, and leftest peers are (56.13, 32.91), (8.25, 23.98), (56.70, 32.44), and (7.85, 24.34), respectively. This means that although the BLACE could shorten the chunk delay and

enhance the content distribution performance of the whole P2P system, it still has room to improve its punishment mechanism, which is used to let a peer get more when it contributes more as well as a peer receive less while it gives less. The reason is that in Fig. 7, the most right (or highest) peer is about 7.22 (or 6.81) times the contribution of the leftest (or lowest) peer, but the proportion of their chunk receiving rates is only 1.33 (or 1.37).

#### IV. CONCLUSION AND FUTURE WORKS

In this article, a content distribution mechanism, BLACE, is proposed to enhance the performance of a mesh P2P streaming system by estimating every peer's contribution with its upload bandwidth, the link latency, and the maximum playing delay. The model and equations of the contribution are also developed to prove the feasibility of BLACE. Moreover, experiment results show that the BLACE could shorten the receiving time and improve the content distribution efficiency of the whole P2P streaming system when it compares with the Coolstreaming, the Coolstreaming<sup>BP</sup>, and the Coolstreaming<sup>LP</sup>. However, the BLACE cannot effectively award more chunks to the peer who contributes more, and punish the peer for giving less and its self-centeredness. Hence, the punishment mechanism will be an issue for further study in the future.

#### REFERENCES

- [1] D.-A. Tran, K.-A. Hua, and T. Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," in *Proc. IEEE Conf. Comput. Commun.*, Mar. 2003, pp. 1283–1292.
- [2] S. Xie, B. Li, G.-Y. Keting, and X. Zhang, "Coolstreaming: Design, theory, and practice," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1661–1671, Dec. 2007.
- [3] Y.-E. Sung, M.-A. Bishop, and S.-G. Rao, "Enabling contribution awareness in an overlay broadcasting system," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1605–1620, Dec. 2007.
- [4] H.-I. Liu and C.-J. Hsu, "CAPS: A contribution-aware P2P streaming system," in *Proc. ISCT*, Oct. 2010, pp. 255–260.
- [5] Y.-W. Chan and Y.-C. Chung, "On the design of contribution-aware p2p streaming networks," in *Proc. JCPC*, Dec. 2009, pp. 805–810.
- [6] H. Chen, C. Liu, and D. Ye, "Contribution-aware overlay optimization for mesh-based live streaming system," in *Proc. 3rd Int. Conf. ASID Commun.*, Aug. 2009, pp. 457–462.
- [7] H.-I. Liu and I.-F. Wu, "MeTree: A contribution and locality-aware P2P live streaming architecture," in *Proc. IEEE Int. Conf. AINA*, Apr. 2010, pp. 1136–1143.
- [8] C.-W. Lo, C.-W. Lin, Y.-C. Chen, and J.-Y. Yu, "Contribution-guided peer selection for reliable peer-to-peer video streaming over mesh networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 9, pp. 1388–1401, Sep. 2012.
- [9] X. Hei, Y. Liu, and K. Ross, "IPTV over P2P streaming networks: The mesh-pull approach," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 86–92, Feb. 2008.
- [10] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proc. IEEE Conf. Comput. Commun.*, May 2007, pp. 919–927.
- [11] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or multiple-tree: A comparative study of live P2P streaming approaches," in *Proc. IEEE Conf. Comput. Commun.*, May 2007, pp. 1424–1432.
- [12] Z. Li and H. Shen, "Game-theoretic analysis of cooperation incentive strategies in mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 8, pp. 1287–1303, Aug. 2012.
- [13] R. Gupta, and Y. N. Singh, "Trust Estimation and Aggregation in Peer-to-Peer Network Using Differential Gossip Algorithm," Jan. 2014 [Online]. Available: <http://arxiv-web3.library.cornell.edu/abs/1210.4301v4>
- [14] The Network Simulator-ns-2. [Online]. Available: <http://www.isi.edu/nsnam/ns>
- [15] S. Naicken, A. Basu, B. Livingston, S. Rodhethbai, and I. Wakeman, "Towards Yet Another Peer-to-Peer Simulator," Apr. 2012 [Online]. Available: [TowardsYetAnotherPeer-to-PeerSimulator](http://TowardsYetAnotherPeer-to-PeerSimulator)
- [16] PlanetLab. [Online]. Available: <http://www.planet-lab.org>
- [17] I. Baumgart, B. Heep, C. Hubsch, and A. Brocco, "OverArch: A common architecture for structured and unstructured overlay networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Orlando, FL, USA, Mar. 2012, pp. 19–24.