

# Contribution-Enabled Congestion-Aware Scheme for P2P Real-Time Multimedia Streaming Systems

Cheng-Yun Ho, Ming-Hsiang Huang, and Chien-Chao Tseng

**Abstract**—In this article, a contribution-enabled congestion-aware (CECA) scheme is proposed to enhance the fairness and efficiency of content distribution, and improve the entire bandwidth utility for mesh peer-to-peer (P2P) real-time multimedia streaming systems. 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 that malfunctions the reward and punishment scheme. Besides, their congestion controls partially or mostly rely on the transport layer protocol such as transmission control protocol (TCP). Hence, the CECA dynamically adjusts a receiver peer's content window size (CWS) based on the contribution and packet loss rate of the receiver peer. Furthermore, the CECA is implemented in NS2 and experimental results show that the CECA could at the most shorten the receiving time about 20.78%, 32.26%, 30.74%, and 26.18%, and improve the entire performance about 11.42%, 18.87%, 14.55%, and 14.19% at the most, when it compares with the BLACE, the Coolstreaming, the Coolstreaming with bandwidth preference, and the Coolstreaming with latency preference, respectively.

**Index Terms**—Contribution, congestion, contribution-enabled, congestion-aware, CECA, 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. File sharing and real-time multimedia streaming are common applications used in P2P networks. 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 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 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 Coolstreaming [2], a sub-stream based mesh P2P streaming system, 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 major factor to solve the mentioned problems and adjust the speed of content distribution. 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. Therefore, BLACE [12] proposed 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 can be a possible contributor before the peer transmits contents to other peers.

Unfortunately, neither Coolstreaming nor BLACE focus on how to award more contents to the peer who distributes more, and punish the peer for its self-centeredness. Furthermore, both Coolstreaming's and BLACE's congestion controls rely on the transmission control protocol (TCP). This means that when the congestion occurs or a packet is lost, TCP will handle the corresponding actions so that Coolstreaming and BLACE neither know nor sense what happened in the network and would keep their content sending rates. Thus, many contents may be temporarily queued in a TCP buffer that will result in a longer delay and a lower throughput. In short, fairness and congestion unawareness are two issues of Coolstreaming and BLACE. Hence, in this article, a contribution-enabled congestion-aware (CECA) scheme is proposed to solve these two problems by dynamically adjusting a receiver peer's content window size (CWS) per

Manuscript received February 5, 2015; revised October 10, 2015.

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, cctsg@cs.nctu.edu.tw).

round-trip time (RTT), equaling a content sending rate from the source peer to the receiver peer, based on the contribution and congestion status (e.g., packet loss rate) of the receiver peer.

The rest of the article is organized as follows. The proposed mechanism and its analysis are 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. CONTRIBUTION-ENABLED CONGESTION-AWARE SCHEME

The core idea of CECA is that a peer contributing more should be awarded more chunks, and on the other hand, the one giving less should be punished for its self-centeredness. In this section, the CECA's contribution-enabled part, the congestion-aware part, and the mathematical analysis are respectively described in the following sub-sections. Besides, the contribution-enabled part is in charge of the peer selection and the CWS increase while the congestion-aware part reacts to the congestion and decreases the CWS.

### A. Contribution-Enabled Part of CECA

In CECA, it is assumed that every peer exchanges its upload bandwidth, link latency, and contribution information with its neighbors periodically. In practice, this assumption is similar to the information sharing between nodes in a reputation system [13], [14] and can be realized by using the methods in [3], [13], [14].

#### 1) Peer selection

The concept of this part is that before a peer including the initiator distributes a chunk to other peers, it needs to calculate how many receiver peers it can support based on its available upload bandwidth, collects its neighbors' contributions, selects the receiver peers with larger contributions from the neighbors, and adjusts the CWSs of the selected receiver peers per RTT. The contribution of a peer  $P$  for a chunk means that the number of peers which 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.

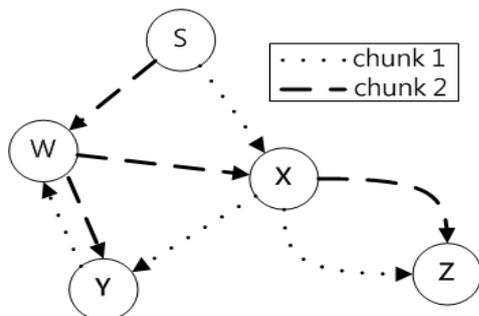


Fig. 1. Example of the contribution of peer  $X$ .

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 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 CECA 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 the CECA has done its work for this chunk.

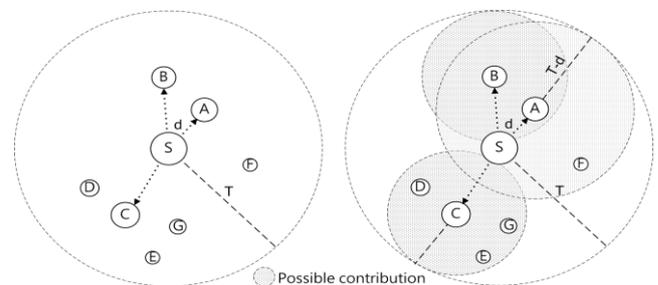


Fig. 2. A case of contribution estimation by peer  $S$ .

#### 2) CWS increase

The way for CWS adjustment is simple and its progresses are as follows. 1) Sort all selected receiver peers by contributions (e.g., from the smallest one to the largest one); 2) Label them and their contributions as  $R_1, R_2, \dots, R_n$  and  $C_1, C_2, \dots, C_n$  based on the result of step 1, respectively; and 3)

Use equation (1) to determine  $R_i$ 's increase amount,  $INC_i$ .

$$INC_i(t) = \begin{cases} 0, & \text{if } CWS_i(t-1) = \max CWS \\ \lceil \overline{RB}(t-1) \times \bar{P}_i(t-1) \rceil \times RTT_i(t-1), & \text{otherwise} \end{cases} \quad (1)$$

where  $\max CWS$  is calculated by the bit-rate of sub-stream (or stream), chunk size, and chunk length,  $t$  means time,  $\overline{RB}$  is the room between the available upload bandwidth (AUB) of source peer and the sum of all CWSs,  $P_i$  is the percentage of  $R_i$ 's contribution (i.e.,  $C_i$ ) to the sum of all receiver peers' contributions, and  $RTT_i$  is the round-trip time of the connection between the source peer and  $R_i$ . Equations (2) and (3) show the details of  $\overline{RB}$  and  $\bar{P}_i$ , respectively. Moreover,  $CWS_i(0)$  equals  $\min CWS$ , whose value is 2;  $P_i(0)$  equals  $1/n$  since, in the initial stage, all peers' contributions are the same; and  $CWS_i(t)$  equals  $\max CWS$  when  $CWS_i(t-1)$  plus  $INC_i(t)$  is larger than  $\max CWS$ .

$$\overline{RB}(t) = AUB(t) - \sum_{i=1}^n CWS_i(t) \quad (2)$$

$$\bar{P}_i(t) = C_i(t) / \sum_{i=1}^n C_i(t) \quad (3)$$

Fig. 3 exhibits the relationship between two CWSs with different contributions when no congestion occurs. From Fig. 3, the CWS of the peer with a larger contribution increases more than the other one with a smaller contribution. In the end, when the two peers arrive at a balance situation, the peer with a larger contribution will have a larger CWS and more contents.

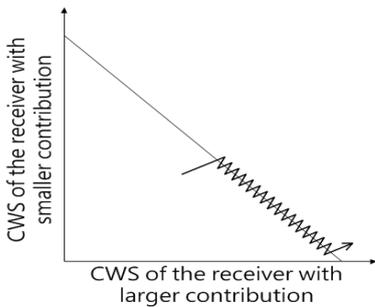


Fig. 3. Case of two CWSs with different contributions.

### B. Congestion-Aware Part of CECA

Similar to the TCP, CECA uses acknowledgements (ACKs) to detect the packet loss between the source peer and receiver peer, and reduces CWS when the congestion occurs. However, unlike TCP, the method of decreasing CWS is shown as equation (4).

$$DEC_i(t) = \begin{cases} 0, & \text{if } CWS_i(t-1) = \min CWS \\ \lceil \overline{RB}(t-1) \times \bar{P}_i(t-1) \rceil \times RTT_i(t-1), & \text{otherwise} \end{cases} \quad (4)$$

where  $DEC_i$  is the decrease amount of the  $CWS_i$  of the receiver peer  $R_i$ ,  $\overline{RB}$  denotes the excess usage of upload bandwidth (i.e., the difference between the sum of all CWSs and the available upload bandwidth) of the source peer, and  $\bar{P}_i$  is the percentage of the reciprocal of  $R_i$ 's contribution (i.e.,

$1/C_i$ ) to the sum of all reciprocals of receiver peers' contributions. Equations (5) and (6) present the details of  $\overline{RB}$  and  $\bar{P}_i$ , respectively. In addition,  $CWS_i(t)$  equals  $\min CWS$  when  $CWS_i(t-1)$  minus  $DEC_i(t)$  is smaller than  $\min CWS$ , and the order of the receiver peers is exactly same as that in the contribution-enabled part.

$$\overline{RB}(t) = \left( \sum_{i=1}^n CWS_i(t) \right) - AUB(t) \quad (5)$$

$$\bar{P}_i(t) = 1 / \left( C_i(t) \times \sum_{i=1}^n \frac{1}{C_i(t)} \right) \quad (6)$$

The reasons for designing this kind of method are that, if the TCP's decreasing way is used, 1) the CWS of the receiver peer with a larger contribution will be reduced more times than that with a smaller contribution, 2) the time for the receiver peer with a larger contribution to recover its CWS would be longer than that with a smaller contribution, and 3) the source peer cannot make significant rewards or punishments to the receiver peers with different contributions.

Fig. 4 demonstrates an example of CWS adjustment flow. In Fig. 4, the source peer knows that the 2<sup>nd</sup> packet (no. 2) is lost according to the ACK at the 11<sup>th</sup> second. Thus, the source peer retransmits the 2<sup>nd</sup> packet and decreases the CWS of the corresponding receiver peer by the equation (4). Similarly, at the 15<sup>th</sup> second, the source peer knows that the 4<sup>th</sup> packet (no. 4) is lost. Then, the source peer retransmits the 4<sup>th</sup> packet, but it does not decrease the CWS this time. This is because the 4<sup>th</sup> packet is sent at the 6<sup>th</sup> second, which is before the last time that the CWS is decreased (at the 11<sup>th</sup> second). Although the ACK received at the 15<sup>th</sup> second contains the lost information of the 2<sup>nd</sup> packet, the source peer will not retransmit the 2<sup>nd</sup> packet. This is because this ACK is in reply to the 5<sup>th</sup> packet (no. 5), which was sent at the 8<sup>th</sup> second, earlier than the time (the 11<sup>th</sup> second) to retransmit the 2<sup>nd</sup> packet. In other words, this ACK cannot indicate the retransmitted 2<sup>nd</sup> packet is lost. Different to the case at the 15<sup>th</sup> second, at the 19<sup>th</sup> second, the ACK still contains the lost information of the 2<sup>nd</sup> packet, so the source peer knows that the retransmitted 2<sup>nd</sup> packet is lost again. Therefore, the source peer retransmits the 2<sup>nd</sup> packet and decreases the CWS again.

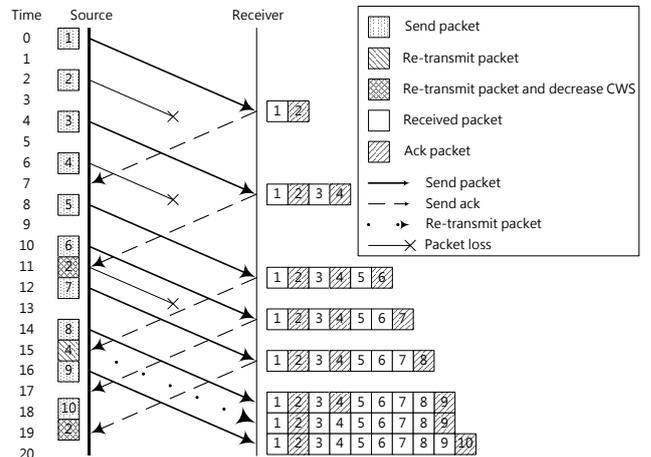


Fig. 4. Example of congestion-aware adjustment flow.

### C. Mathematical Analysis of CECA

#### 1) More contributions more rewards

In the contribution-enabled part, the  $CWS_i$  increases at a rate of  $INC_i/RTT_i$  (equaling  $\lceil RB(t-1) \times P_i(t-1) \rceil$ ) per second, for every  $R_i$ . For any  $R_i$  and  $R_j$ , suppose that  $C_i \leq C_j$ . The increase rates of  $R_i$  and  $R_j$  are  $INC_i/RTT_i$  and  $INC_j/RTT_j$ , respectively. From the equation (7), the increase rate of  $CWS_i$  is not faster than that of  $CWS_j$ . Hence, more contributions more rewards that is proved.

$$\frac{INC_i/RTT_i}{INC_j/RTT_j} = \frac{\lceil RB(t-1) \times P_i(t-1) \rceil}{\lceil RB(t-1) \times P_j(t-1) \rceil} \leq \frac{\lceil RB(t-1) \times P_j(t-1) \rceil}{\lceil RB(t-1) \times P_j(t-1) \rceil} = 1 \quad (7)$$

$\because C_i \leq C_j \Rightarrow C_i / \sum_{k=1}^n C_k \leq C_j / \sum_{k=1}^n C_k \Rightarrow P_i \leq P_j$  from equ. (3)  $\Rightarrow RB \times P_i \leq RB \times P_j \Rightarrow \lceil RB \times P_i \rceil \leq \lceil RB \times P_j \rceil$

#### 2) Less contributions more penalties

Similarly, in the congestion-aware part, the  $CWS_i$  decreases at a rate of  $DEC_i/RTT_i$  (equaling  $\lceil \overline{RB}(t-1) \times \overline{P}_i(t-1) \rceil$ ) per second, for every  $R_i$ . When  $C_i \leq C_j$ , for any  $R_i$  and  $R_j$ , the decrease rate of  $CWS_i$  is not slower than that of  $CWS_j$  based on the equation (8). Hence, it proves that less contributions more penalties.

$$\frac{DEC_i/RTT_i}{DEC_j/RTT_j} = \frac{\lceil \overline{RB}(t-1) \times \overline{P}_i(t-1) \rceil}{\lceil \overline{RB}(t-1) \times \overline{P}_j(t-1) \rceil} \geq \frac{\lceil \overline{RB}(t-1) \times \overline{P}_j(t-1) \rceil}{\lceil \overline{RB}(t-1) \times \overline{P}_j(t-1) \rceil} = 1 \quad (8)$$

$$\because C_i \leq C_j \Rightarrow 1/C_i \geq 1/C_j \Rightarrow \frac{1}{C_i} \times \frac{1}{\sum_{k=1}^n C_k} \geq \frac{1}{C_j} \times \frac{1}{\sum_{k=1}^n C_k} \Rightarrow \overline{P}_i \geq \overline{P}_j$$

from equ.(6)  $\Rightarrow \overline{RB} \times \overline{P}_i \geq \overline{RB} \times \overline{P}_j \Rightarrow \lceil \overline{RB} \times \overline{P}_i \rceil \geq \lceil \overline{RB} \times \overline{P}_j \rceil$ .

#### 3) Bound for every second

1) In the contribution-enabled part, for every second, the sum of increments can be represented by  $\sum INC_i/RTT_i$ .

Moreover,

$$\begin{aligned} \sum_{i=1}^n \frac{INC_i}{RTT_i} &= \sum_{i=1}^n \lceil RB \times P_i \rceil \leq \sum_{i=1}^n (RB \times P_i + 1) = \sum_{i=1}^n RB \times P_i + \sum_{i=1}^n 1 \\ &= RB \times \sum_{i=1}^n P_i + n = RB + n \quad (\because \forall \text{ second, } RB \text{ is constant and } \sum_{i=1}^n P_i = 1). \end{aligned}$$

Hence,  $RB+n$  is the upper bound in the contribution-enabled part.

2) In the congestion-aware part, for every second, the sum of decrements can be represented by  $\sum DEC_i/RTT_i$ .

Moreover,

$$\begin{aligned} \sum_{i=1}^n \frac{DEC_i}{RTT_i} &= \sum_{i=1}^n \lceil \overline{RB} \times \overline{P}_i \rceil \geq \sum_{i=1}^n (\overline{RB} \times \overline{P}_i - 1) = \sum_{i=1}^n \overline{RB} \times \overline{P}_i - \sum_{i=1}^n 1 \\ &= \overline{RB} \times \sum_{i=1}^n \overline{P}_i - n = \overline{RB} - n \quad (\because \forall \text{ second, } \overline{RB} \text{ is constant and } \sum_{i=1}^n \overline{P}_i = 1). \end{aligned}$$

Hence,  $\overline{RB}-n$  is the lower bound in the congestion-aware part.

From both 1) and 2), bound for every second is proved.

### III. PERFORMANCE EVALUATIONS

To observe how much performance the CECA can improve in a P2P streaming system, we chose NS2 [15] as a simulation tool and executed a lot of experiments to compare the CECA with the BLACE, Coolstreaming, Coolstreaming with bandwidth preference (Coolstreaming<sup>BP</sup>), and Coolstreaming with latency preference (Coolstreaming<sup>LP</sup>), where the Coolstreaming<sup>BP</sup> and Coolstreaming<sup>LP</sup> respectively denote the Coolstreaming uses the upload bandwidth and link latency as the only factor to select receivers. Furthermore, the Coolstreaming<sup>BP</sup> and Coolstreaming<sup>LP</sup> can be treated as other P2P streaming systems. The reasons for choosing NS2, BLACE, and Coolstreaming are as follows. 1) S. Naicken *et al.* [16] investigated many existing P2P simulators, such as PlanetLab and OverSim, 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 network simulator in the academic community; 4) BLACE is a recent P2P mechanism using contributions to evaluate and select peers to accelerate the content distribution; and 5) 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

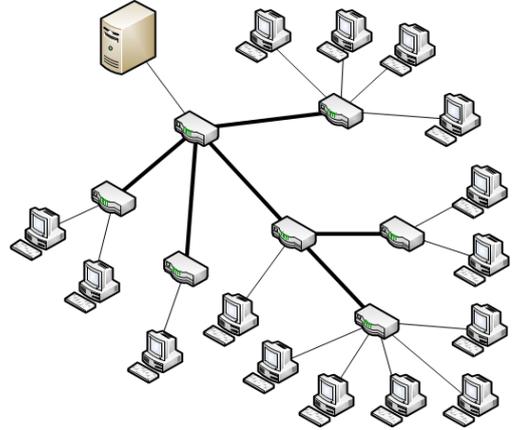


Fig. 5. A diagram of network topologies.

TABLE I: 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	Type 1: bandwidth of edge link	800-1200 Kbps (up)/ 500 Kbps (down)
Type 2: bandwidth of edge link	400-750 Kbps (up)/ 500 Kbps (down)	Type 3: bandwidth of edge link	100-300 Kbps (up)/ 500 Kbps (down)
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 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 receiver peers) and follows a power-law degree structure [17], [18]. The routers and the peers present the cores and the edges of the network, respectively. Furthermore, the link between two routers is called core link while the link between the peer and the router is called edge link. All links' latencies are from 0.001 to 0.1 seconds and all core links' bandwidths (full duplex) are from 100 to 1000 Mbps. The bandwidth of the edge link (full duplex) between the source peer to the router is 5 Mbps while the other edge links' download bandwidths are 500 Kbps and upload bandwidths are dispatched to three types: 1) from 800 to 1200 Kbps, 2) from 400 to 750 Kbps, and 3) from 100 to 300 Kbps. This is because when the source peer supports more receivers and the receiver peers have more different contributions, the CECA can more obviously demonstrate its characteristics and strengths. More parameter settings are illustrated in Table I.

**B. Experiment Results of Chunk Delay and Receiving Rate**

Fig. 6 and Fig. 7 exhibit the experiment results of five content distribution mechanisms with variable joining/leaving peers. Note the five mechanisms are CECA, BLACE, Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup>. Furthermore, the peer joining/leaving rate (JLR) of every experiment is fixed (e.g., 1, 2, 3, and 4 per 10 seconds).

In Fig. 6, compared with the BLACE, Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup>, the CECA shortens the chunk delay (receiving time) about 19.53%, 31.71%, 30.74%, and 26.18%, respectively, 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 CECA still has the littlest chunk delay among these five mechanisms. For example, while the peer JLR is 4 peers per 10 seconds, the chunk delay of CECA, BLACE, Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup> are 7.45, 9.41, 11.0, 10.72, and 9.77 seconds.

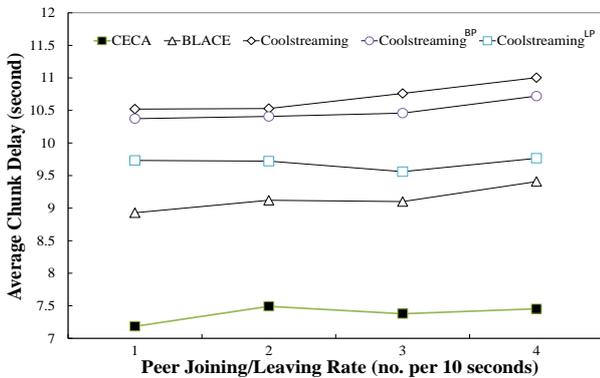


Fig. 6. Average chunk delay with different peer JLRs.

On the other point of view, as shown in Fig. 7, the CECA has better performance for every sub-stream than others. For example, compared with the BLACE, the CECA improves the average chunk receiving rate about 8.58%, 9.03%, 10.61%, and 11.42% when the peer JLR is from 1 to 4 peers per 10 seconds, respectively. Similarly, CECA improves the average

chunk receiving rate of Coolstreaming about 15.51%, 12.24%, 18.41%, and 18.87%, respectively. Even in the worst case (i.e., 2 peers join/leave the network every 10 seconds), for every sub-stream, the CECA respectively achieves 1.09, 1.12, 1.10, and 1.11 times higher average chunk receiving rate in comparison with BLACE, Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup>. Hence, the CECA can make the entire P2P system more efficient.

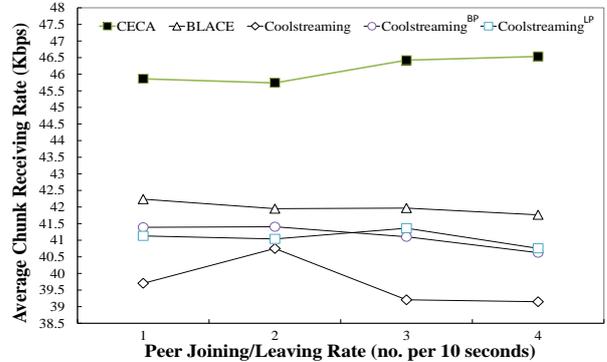


Fig. 7. Average chunk receiving rate with different peer JLRs.

**C. Experiment Results of Fairness and Bandwidth Utility**

To evaluate the fairness of the receiver peers in a P2P system, a fairness index function (9) [19] is used in this subsection.

$$F(x) = \frac{(\sum x_i)^2}{n(\sum x_i^2)} \tag{9}$$

where  $n$  equals the number of receiver peers and

$$x_i = \begin{cases} 1, & \text{if } \gamma_i = \text{bit-rate of stream and } \zeta_i \geq \text{bit-rate of stream} \\ \gamma_i / \zeta_i, & \text{otherwise} \end{cases}$$

where  $\gamma_i$  and  $\zeta_i$  respectively mean that the chunk receiving rate and chunk sending rate of the receiver peer  $i$  for a stream. In this article,  $n$  is 1000 and the bit-rate of stream is 384 Kbps.

The result of fairness index function ranges from  $1/n$  (worst case) to 1 (best case). Moreover, the result will be the maximum if all receiver peers have the same ratio of “how many chunks they get” to “how many chunks they contribute.” In other words, when a content distribution mechanism has a well punishment method, it will let a peer get more when contributing more as well as a peer receive less while giving less, and the result of fairness index function tends towards 1.

Table II shows the fairness index results of CECA, BLACE, Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup>. In Table II, the CECA has the highest fairness index value among the five content distribution mechanisms. For example, the CECA increases the fairness index of the BLACE and Coolstreaming about 0.92% and 0.59%, respectively.

On the other point of view, as shown in Table III, the CECA can more efficiently use the upload bandwidth than the others. In practice, compared with the Coolstreaming, the CECA improves the upload bandwidth utility about 17.91% while the CECA adds the upload bandwidth utility of the BLACE about 9.61%.

TABLE II: FAIRNESS INDEX OF DIFFERENT CONTENT DISTRIBUTION MECHANISMS

Content Distribution Mechanisms				
CECA	BLACE	Coolstreaming	Coolstreaming <sup>BP</sup>	Coolstreaming <sup>LP</sup>
96.48%	95.56%	95.89%	95.76%	95.64%

TABLE III: BANDWIDTH UTILITY OF DIFFERENT CONTENT DISTRIBUTION MECHANISMS

Content Distribution Mechanisms				
CECA	BLACE	Coolstreaming	Coolstreaming <sup>BP</sup>	Coolstreaming <sup>LP</sup>
54.22%	44.61%	36.31%	38.99%	40.27%

#### IV. CONCLUSIONS AND FUTURE WORKS

In this article, a content distribution mechanism, CECA, is proposed to enhance the fairness and efficiency of content distribution, and improve the entire bandwidth utility for a mesh P2P real-time multimedia streaming system by dynamically adjusting a receiver peer's content window size according to the contribution and packet loss rate of the receiver peer. Experiment results show that the CECA could really shorten the receiving time, enhance the reward and punishment scheme, and improve the bandwidth utility and the content distribution efficiency of the entire P2P streaming system when it compares with the BLACE, Coolstreaming, Coolstreaming<sup>BP</sup>, and Coolstreaming<sup>LP</sup>. However, the CECA cannot fully use the available bandwidth to make every receiver peer have a 100% of stream. Therefore, how to solve the problems to reach an ideal objective 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. the IEEE Conference on Computer Communications*, March 2003, pp. 1283-1292.
- [2] S. Xie, B. Li, G.-Y. Keting, and X. Zhang, "Coolstreaming: Design, theory, and practice," *IEEE Transactions on Multimedia*, vol. 9, issue 8, pp. 1661-1671, December 2007.
- [3] Y.-E. Sung, M.-A. Bishop, and S.-G. Rao, "Enabling contribution awareness in an overlay broadcasting system," *IEEE Transactions on Multimedia*, vol. 9, issue 8, pp. 1605-1620, December 2007.
- [4] H.-I. Liu and C.-J. Hsu, "CAPS: A contribution-aware P2P streaming system," in *Proc. International Symposium on Communications and Information Technologies*, October 2010, pp. 255-260.
- [5] Y.-W. Chan and Y.-C. Chung, "On the design of contribution-aware P2P streaming networks," in *Proc. the Joint Conferences on Pervasive Computing*, December 2009, pp. 805-810.
- [6] H. Chen, C. Liu, and D. Ye, "Contribution-aware overlay optimization for mesh-based live streaming system," in *Proc. the third International Conference on Anti-counterfeiting, Security, and Identification in Communication*, August 2009, pp. 457-462.
- [7] H.-I. Liu and I.-F. Wu, "MeTree: A contribution and locality-aware P2P live streaming architecture," in *Proc. the IEEE International Conference on Advanced Information Networking and Applications*, April 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 Transactions on Circuits and Systems for Video Technology*, vol. 22, issue 9, pp. 1388-1401, September, 2012.

- [9] X. Hei, Y. Liu, and K. Ross, "IPTV over P2P streaming networks: The mesh-pull approach," *IEEE Communications Magazine*, vol. 46, issue 2, pp. 86-92, February 2008.
- [10] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proc. the IEEE Conference on Computer Communications*, 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. the IEEE Conf. on Computer Communications*, 2007, pp. 1424-1432.
- [12] C.-Y. Ho, M.-H. Huang, C.-Y. Ho, and C.-C. Tseng, "Bandwidth and latency aware contribution estimation in P2P streaming system," *IEEE Communications Letters*, pp. 1511-1514, September 2014.
- [13] Z. Li and H. Shen, "Game-theoretic analysis of cooperation incentive strategies in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 11, issue 8, pp. 1287-1303, August 2012.
- [14] R. Gupta and Y. N. Singh. (Jan. 2014). Trust estimation and aggregation in peer-to-peer network using differential gossip algorithm. [Online]. Available: <http://arxiv-web3.library.cornell.edu/abs/1210.4301v4>
- [15] The network simulator — ns-2. [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [16] S. Naicken, A. Basu, B. Livingston, S. Rodhethbai, and I. Wakeman. (Apr. 2012). Towards yet another peer-to-peer simulator. [Online]. Available: <http://sro.sussex.ac.uk/37851/>
- [17] A. Mahanti, N. Carlsson, A. Mahanti, M. Arlitt, and C. Williamson, "A tale of the tails: Power-laws in internet measurements," *IEEE Network*, vol. 27, issue 1, pp. 59-64, January-February, 2013.
- [18] A. Hussain, K. Latif, A. T. Rextin, A. Hayat, and M. Alam, "Scalable visualization of semantic nets using power-law graphs," *Applied Mathematics & Information Sciences*, vol. 8, no. 1, pp. 355-367, January 2014.
- [19] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report TR-301, 1984.



**Cheng-Yun Ho** received his B.S. degree in computer science from National Cheng Chi University, Taiwan, in 2007, the M.S. degree in networking engineering from National Chiao Tung University, Taiwan, in 2008, then enrolled into Ph.D. program directly. He is currently pursuing the Ph.D. degree in computer science and engineering at National Chiao Tung University, Hsinchu, Taiwan. His research interests include peer-to-peer (P2P) network, information-centric networking (ICN), computer networks (including wireless and mobile networks), and network protocols.



**Ming-Hsiang Huang** received his B.S., and M.S. degrees in computer science and engineering from National Chiao Tung University, Taiwan in 2008, and 2010 respectively. He is currently pursuing the Ph.D. degree at the Institute of Computer Science and Engineering of National Chiao Tung University. His research interest is in compilation techniques on heterogeneous platforms.



**Chien-Chao Tseng** received his B.S. degree in industrial engineering from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981; M.S. and Ph.D. degrees in computer science from the Southern Methodist University, Dallas, Texas, USA, in 1986 and 1989, respectively. He is currently a professor in the Department of Computer Science at National Chiao-Tung University, Hsin-Chu, Taiwan. His research interests include wireless internet, heterogeneous networks, and software defined networks.