PAPER Special Section on Parallel and Distributed Computing and Networking

Traffic Engineering of Peer-Assisted Content Delivery Network with Content-Oriented Incentive Mechanism

Naoya MAKI[†a)], Takayuki NISHIO[†b)], *Student Members*, Ryoichi SHINKUMA[†c)], *Senior Member*, Tatsuya MORI[††d)], Noriaki KAMIYAMA[††e)], Ryoichi KAWAHARA[††f)], *Members*, *and* Tatsuro TAKAHASHI[†g)], *Fellow*

SUMMARY In content services where people purchase and download large-volume contents, minimizing network traffic is crucial for the service provider and the network operator since they want to lower the cost charged for bandwidth and the cost for network infrastructure, respectively. Traffic localization is an effective way of reducing network traffic. Network traffic is localized when a client can obtain the requested content files from other a near-by altruistic client instead of the source servers. The concept of the peer-assisted content distribution network (CDN) can reduce the overall traffic with this mechanism and enable service providers to minimize traffic without deploying or borrowing distributed storage. To localize traffic effectively, content files that are likely to be requested by many clients should be cached locally. This paper presents a novel traffic engineering scheme for peer-assisted CDN models. Its key idea is to control the behavior of clients by using content-oriented incentive mechanism. This approach enables us to optimize traffic flows by letting altruistic clients download content files that are most likely contributed to localizing traffic among clients. In order to let altruistic clients request the desired files, we combine content files while keeping the price equal to the one for a single content. This paper presents a solution for optimizing the selection of content files to be combined so that cross traffic in a network is minimized. We also give a model for analyzing the upper-bound performance and the numerical results.
key words: *content delivery network, peer-assisted network, contents combinations, combining contents, traffic localization*

1. Introduction

Content services have been in the mainstay of the Internet market for the last ten years; people purchase music, movies, and application software by the Internet. Since the volumes of such files tend to be quite large, it is important to minimize the amount of traffic in their transactions; the content service provider and the network operator want to minimize the cost charged for bandwidth and the cost for network infrastructure, respectively. Localization effectively reduces traffic; if a content file is stored at multiple locations, downloading from the nearest server generally minimizes the bandwidth consumption. In content delivery networks (CDNs), which are distributed storage networks that achieve traffic localization; in a CDN, a content request from a client is directed to the replica nearest the client [1]–[3]. CDNs have been extensively used. However, from the service-provider viewpoint, CDNs are not always profitable because it costs a lot to deploy and maintain distributed storages or to rent them. Therefore, peer-assisted CDNs have been proposed as a way of overcoming these drawbacks [4]–[7]. In a peer-assisted CDN, a client request is directed to the nearest replica as in normal CDNs, but the replica can be in the cache of one of millions of clients. Peer-assisted CDNs allow service providers to minimize traffic without deploying or borrowing distributed storages.

In peer-assisted CDNs, clients are expected to contribute to uploading their cached content files to other geolocally close clients. However, as reported in [8], [9], most clients in P2P applications have non-altruistic attitudes concerning the services. Although much research have addressed incentive mechanisms that encourage such clients to contribute to services, expecting this in peer-assisted CDNs is unlikely because those clients are customers of the service providers. Therefore, we can expect a limited number of clients to altruistically contribute to the services [10], [11]. We should consider how effectively we utilize the limited cache spaces of the altruistic clients to achieve traffic localization.

Our goal is to reduce cross traffic among clients in peer-assisted CDNs. One of the simplest ways to achieve the goal is to automatically push content files, which are likely be requested by other clients, to some clients. However, it is not clear that even an altruistic client would be willing to accept caching content files that are completely independent of his/her demand.

Instead of trying to exploit the resources of altruistic clients, we aim to adjust the behavior of altruistic clients by using content-oriented incentive mechanism. This approach enables us to optimize traffic flows by letting altruistic clients download content files that are most likely contribute to localizing traffic among clients. In order to let altruistic clients request the desired files, we combine content files while keeping the price equal to the one for a single content as illustrated in Fig. 1. The advantage of our approach is that we avoid exploiting the valuable resources contributed from the altruistic clients but give them clear incentive so that they will contribute to the system in a sus-

Manuscript received January 6, 2012.
Manuscript revised May 12, 2012.
[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
[††]The authors are with NTT Network Technology Laboratories, NTT Corporation, Musashino-shi, 180–8585 Japan.
a) E-mail: maki@cube.kuee.kyoto-u.ac.jp
b) E-mail: nishio@cube.kuee.kyoto-u.ac.jp
c) E-mail: shinkuma@i.kyoto-u.ac.jp
d) E-mail: mori.tatsuya@lab.ntt.co.jp
e) E-mail: kamiyama.noriaki@lab.ntt.co.jp
f) E-mail: kawahara.ryoichi@lab.ntt.co.jp
g) E-mail: ttakahashi@i.kyoto-u.ac.jp
DOI: 10.1587/transinf.E95.D.2860

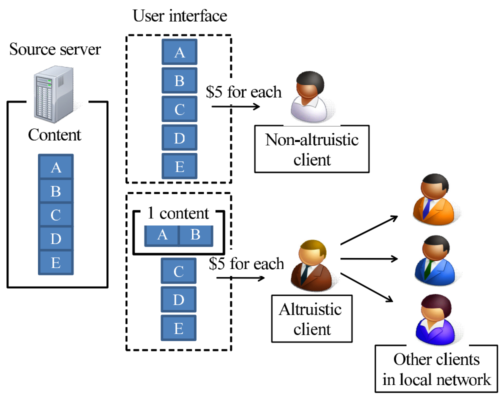

Fig. 1 Example of content combination. A combination of content A and B is available for altruistic client for $5. The combination will likely be requested in the local network.

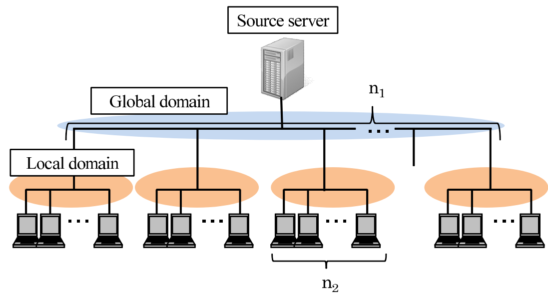

Fig. 2 Network model.

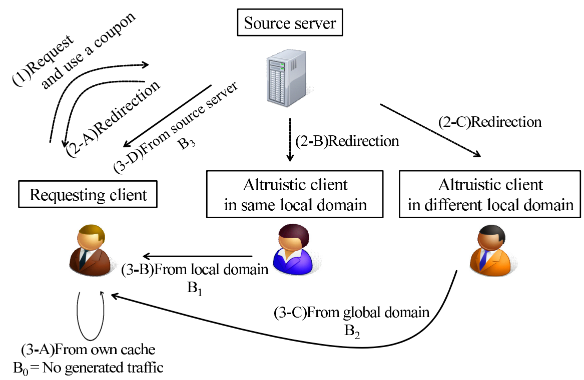

Fig. 3 Peer-assisted CDN model.

tainable way. The content-oriented incentive mechanism is on the basis of a simple assumption that the probability that combined content files, which include a single content file, is always equal to or larger than the probability that the single content files is requested. We note that the recent drastic increase in the capacity of cache equipment for clients such as HDD also supports the assumption. One clear drawback of the approach is that transferring combined content files to altruistic clients could generate a large amount of traffic in a short time period. To address the problem, this paper adjusts the trade-off between the increase in traffic resulting from content combination and the reduction in traffic resulting from traffic localization.

In this paper, we propose a novel traffic engineering scheme based on the above idea. The following are the contributions of our paper: i) we induce altruistic clients to cache files that optimally localize traffic; ii) we solve the optimal selection of combined content files for altruistic clients and describe its algorithms; iii) we show the upper-bound of the performance of traffic localization due to content combination while taking into account the above trade-off.

The rest of this paper is organized as follows. Section 2 describes our assumptions about our network structure and the content download model. Section 3 introduces our architecture and the algorithm that selects content combinations. Section 4 gives our model to analyze the upper-bound performance and numerical results. Section 5 introduces related work and compares our scheme with them. We conclude in Sect. 6.

2. Service Model

In this paper, we assume a content service in which clients are charged at a fixed rate and a fixed number of coupons are given to them at every period. They can purchase a content file or a set of combined content files in exchange for a coupon. Even when clients enjoy their purchased content again, they must use a coupon. The fixed charge is essential in our system so that providing combined content files with the single content price does not reduce revenue of the service provider. We assume the service only deals with such common entertainment as musics and movies. We denote a set of all the content files purchasable in the service as $\mathbb{C}$. We assume that each content has a certain popularity and clients request content based on a probability. Note that it has been well discussed how to estimate content popularity [18] and it should be out of scope in this paper. In our model, the popularity of a content is preliminarily estimated by an existing method and the request probability of a content follows Zipf's law [12]. The request probability of a content which has the i-th highest popularity, P_i, is:

$$P_{\mathrm{i}} = \frac{\frac{1}{i}}{\sum_{j\in\mathbb{C}} \frac{1}{j}}. \tag{1}$$

Figure 2 illustrates our assumed network model. This is a hierarchical model where the local and global domains, and source servers are in the bottom, middle, and top layers, respectively. Figure 3 illustrates a peer-assisted CDN model we assume. (1) The client first makes a request for content and uses a coupon. (2) Then the source server makes a transaction for the content charge and redirects the request to the cached location that minimizes the traffic; the redirection is performed with the following priorities: A) client's own cache, B) cache at other altruistic clients in the local domain to which the requester belongs, C) cache at other altruistic clients in the global domain, D) and the source server. (3) When the requester retrieves the requested content from the redirected location A), B), C) or D), traffic B_0, B_1, B_2, and B_3 are generated ($B_0(=0) < B_1 < B_2 < B_3$).

3. Proposed Scheme

In the following discussions, we assume that the system is time-slotted. At each unit-time, only a client is permitted to request and download a content file. When the client is an altruistic client, the content file could be a set of combined files, which are chosen based on the algorithms described in Sect. 3.2. Thus, it does not happen that multiple altruistic clients request combined content files simultaneously. Even when our method is used in real peer-assisted CDNs, combined content files are not provided for multiple altruistic clients, which can be done because all client requests are handled by the service provider in the centralized manner in peer-assisted CDNs.

The unit time can be considered as the average interval of client requests because, as we mentioned above, a client makes a request at each unit-time. In the following discussions, to simplify the model, we assumed coupons are provided frequently enough compared with the average interval of client requests; we do not assume that a client cannot make a request because s/he runs out of coupons. Even if we consider the case that clients run out of coupons, it would not change our conclusion in the paper because it does not affect how to choose combined content files and how much traffic is reduced in the proposed scheme and it affects equally on the conventional and proposed schemes.

3.1 Incentive Mechanism

As mentioned in Sect. 1, our key idea is content combination that induces altruistic clients to cache content files that are likely to be requested in local networks. In this paper, we simply consider the request probability of a set of combined content files, P_{comb}, to be:

$$P_{\mathrm{comb}} = \sum_{i \in \mathbb{B}} P_{\mathrm{i}}, \tag{2}$$

where $\mathbb{B}$ is a set of the combined content files. Figure 4 illustrates an example of the request probability of the combined content files. The request probability of the combination of contents B and D equals to the sum of the request probabilities of those files.

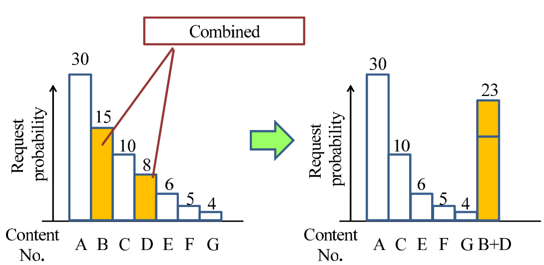

Fig. 4 Example of request probability of combined content. Content files B and D are combined.

3.2 Optimal Selection of Combined Content Files

3.2.1 Problem Formulation

Let us here discuss the optimal selection of combined content files. Suppose that altruistic client u is going to request a content file at time t. Altruistic client u newly obtains a bunch of combined content files $\mathbb{B}_{\mathrm{u}}$ and removes a set of content files $\mathbb{D}_{\mathrm{u}}$ from its cache $\mathbb{C}_{\mathrm{u}}^{\mathrm{t}}$ to make space for $\mathbb{B}_{\mathrm{u}}$. To determine the combination of content files, the service provider solves optimization problem written as following:

$$\max_{\mathbb{B}_{\mathrm{u}},\mathbb{D}_{\mathrm{u}}} T(\zeta(S_{\mathrm{t}}) - \zeta(S_{\mathrm{t+1}})) - \eta(\mathbb{B}_{\mathrm{u}}) \tag{3}$$

s.t.

$$\begin{aligned}
&S_{\mathrm{t}} \cap S_{\mathrm{t+1}} = (\mathbb{C}_1, \cdots, \mathbb{C}_{\mathrm{u-1}}, \mathbb{C}_{\mathrm{u+1}} \cdots, \mathbb{C}_{\mathrm{N}})\\
&S_{\mathrm{t}} = \left(\mathbb{C}_1, \cdots, \mathbb{C}_{\mathrm{u}}^{\mathrm{t}}, \cdots, \mathbb{C}_{\mathrm{N}}\right)\\
&S_{\mathrm{t+1}} = \left(\mathbb{C}_1, \cdots, \mathbb{C}_{\mathrm{u}}^{\mathrm{t+1}}, \cdots, \mathbb{C}_{\mathrm{N}}\right)\\
&\mathbb{C}_{\mathrm{u}}^{\mathrm{t+1}} = \mathbb{C}_{\mathrm{u}}^{\mathrm{t}} + \mathbb{B}_{\mathrm{u}} - \mathbb{D}_{\mathrm{u}}
\end{aligned}$$

where N is the total number of clients; S_{t} is the state of the cache in the entire network at time t; $\zeta(S_{\mathrm{t}})$ indicates generated traffic when the cache state is S_{t}; $t+1$ means the time just after the cache of client u is replaced; $\eta(\mathbb{B}_{\mathrm{u}})$ indicates how much traffic is increased by downloading a set of content files $\mathbb{B}_{\mathrm{u}}$ compared with downloading a single file. As we can see from the definition in Eq. (3), the difference between S_{t} and $S_{\mathrm{t+1}}$ is $\mathbb{B}_{\mathrm{u}} - \mathbb{D}_{\mathrm{u}}$. $\zeta(S_{\mathrm{t}}) - \zeta(S_{\mathrm{t+1}})$ is how much traffic will be reduced from t to $t+1$ as a result of caching and discarding $\mathbb{B}_{\mathrm{u}}$ and $\mathbb{D}_{\mathrm{u}}$. If the cached content files at clients in the network do not significantly change during T, $T(\zeta(S_{\mathrm{t}}) - \zeta(S_{\mathrm{t+1}}))$ indicates the amount of reduced traffic during period T. The start time of period T is time $t+1$: a unit time after combined content files are requested and obtained at time t. The end time of period T is $t+1+T$: T unit times after $t+1$. However, as mentioned in Sect. 1, downloading $\mathbb{B}_{\mathrm{u}}$ instantaneously generates a large amount of traffic, which is represented as $\eta(\mathbb{B}_{\mathrm{u}})$ in Eq. (3).

We solve for Eq. (3) and describe it as a set of algorithms in the next section that are roughly split into two phases: analysis and selection.

3.2.2 Analysis Phase

We first find out how much traffic is generated when client v requests and retrieves content i. The generated traffic depends on which cached location content i is retrieved from, client v's cache, the local domain of client v, the global domain of client v, or the source server. In Alg. 1, the generated traffic is represented as B_{vi}. $\mathbb{C}_v^l$ is the set of contents in the cache of the altruistic clients in a local domain that client v has joined and $\mathbb{C}_v^g$ is a set of contents cached by altruistic clients in other local domains.

We estimate how much traffic is expected to be reduced at time $t+1$ if altruistic client u requests and cache content

Algorithm 1 Algorithm for B_{vi}

Require: $\mathbb{C}_v$, $\mathbb{C}_v^l$, $\mathbb{C}_v^g$ are known.
if $i \in \mathbb{C}_v$ **then**
  $B_{\mathrm{vi}} \leftarrow B_0$
else if $i \in \mathbb{C}_v^l$ **then**
  $B_{\mathrm{vi}} \leftarrow B_1$
else if $i \in \mathbb{C}_v^g$ **then**
  $B_{\mathrm{vi}} \leftarrow B_2$
else
  $B_{\mathrm{vi}} \leftarrow B_3$
end if

Algorithm 2 Algorithm for Δ_{j}^-

Require: $\mathbb{U}_v^l$, $\overline{\mathbb{U}_v^l}$, $\mathbb{C}_v$ are known.
for all v (every client) **do**
  $\delta_{\mathrm{vj}}^- \leftarrow 0$
  if v equals u **then**
    $\delta_{\mathrm{vj}}^- \leftarrow B_{\mathrm{vj}} - B_0$
  else if $v \in \mathbb{U}_u^l$ **then**
    if in $\mathbb{U}_v^l$ there is no altruistic client who has content j in his/her cache and $j \notin \mathbb{C}_v$ **then**
      $\delta_{\mathrm{vj}}^- \leftarrow B_{\mathrm{vj}} - B_1$
    end if
  else if $v \in \overline{\mathbb{U}_u^l}$ **then**
    if in $\overline{\mathbb{U}_v^l}$ there is no altruistic client who has content j in his/her cache and $j \notin \mathbb{C}_v$ **then**
      $\delta_{\mathrm{vj}}^- \leftarrow B_3 - B_2$
    end if
  end if
end for
$\Delta_{\mathrm{j}}^- \leftarrow 0$
for all v (every client) **do**
  $\Delta_{\mathrm{j}}^- \leftarrow \Delta_{\mathrm{j}}^- + \frac{P_j \cdot \delta_{\mathrm{vj}}^-}{N}$
end for

Algorithm 3 Algorithm for Δ_{k}^+

Require: $\mathbb{U}_v^l$, $\overline{\mathbb{U}_v^l}$, $\mathbb{C}_v$ are known.
for all v (every client) **do**
  $\delta_{\mathrm{vk}}^+ \leftarrow 0$
  if v equals u (i.e. altruistic client u him/herself) **then**
    if in $\mathbb{U}_u^l$ there is an altruistic client who has content k in his/her cache **then**
      $\delta_{\mathrm{vk}}^+ \leftarrow B_1 - B_0$
    else if in $\overline{\mathbb{U}_u^l}$ there is an altruistic client who has content k in his/her cache **then**
      $\delta_{\mathrm{vk}}^+ \leftarrow B_2 - B_0$
    else
      $\delta_{\mathrm{vk}}^+ \leftarrow B_3 - B_0$
    end if
  else if $v \in \mathbb{U}_u^l$ **then**
    if in $\mathbb{U}_v^l$ there is no altruistic client who has content k in his/her cache and $k \notin \mathbb{C}_v$ **then**
      if in $\overline{\mathbb{U}_v^l}$ there is an altruistic client who has content k in his/her cache **then**
        $\delta_{\mathrm{vk}}^+ \leftarrow B_2 - B_1$
      else
        $\delta_{\mathrm{vk}}^+ \leftarrow B_3 - B_1$
      end if
    end if
  else if $v \in \overline{\mathbb{U}_u^l}$ **then**
    if in $\overline{\mathbb{U}_v^l}$ there is no altruistic client who has content k in his/her cache and $k \notin \mathbb{C}_v$ **then**
      $\delta_{\mathrm{vk}}^+ \leftarrow B_3 - B_2$
    end if
  end if
end for
$\Delta_{\mathrm{k}}^+ \leftarrow 0$
for all v (every client) **do**
  $\Delta_{\mathrm{k}}^+ \leftarrow \Delta_{\mathrm{k}}^+ + \frac{P_k \cdot \delta_{\mathrm{vk}}^+}{N}$
end for

j at time t, which is represented as Δ_{j}^-. Δ_{j}^- is obtained from δ_{vj}^- which is how much generated traffic would be reduced when client v requests and obtains content j at time $t+1$. In Alg. 2, $\mathbb{U}_v^l$ and $\overline{\mathbb{U}_v^l}$ represent a set of clients that have joined the same local domain as that of client v and that of other local domains, respectively.

Algorithm 3 estimates how much traffic is expected to increase at time $t+1$ if altruistic client u removes content k from its cache at time t, which is represented as Δ_{k}^+. Δ_{k}^+ is obtained from δ_{vk}^+ which is how much generated traffic would increase when client v requests and obtains content k at time $t+1$. P_{k} should be also considered to calculate Δ_{k}^+.

3.2.3 Selection Phase

We here discuss how we find the sets of combined and discarded content files $\mathbb{B}_{\mathrm{u}}$ and $\mathbb{D}_{\mathrm{u}}$ that satisfy Eq. (3). First, to optimize $\mathbb{B}_{\mathrm{u}}$, as we mentioned in Sect. 1, we have to consider the fact that, as we increase the number of combined content files, more instantaneous traffic would generate. That is, the expected traffic reduction by content j should be given by:

$$E_{\mathrm{j}} = T \cdot \Delta_{\mathrm{j}}^- - B_{\mathrm{uj}}, \tag{4}$$

where B_{uj} and Δ_{j}^- are obtained by Algs. 1 and 2 and T is defined in Sect. 3.2.1. In addition, P_{comb} defined in Eq. (2) should be also considered because content j would not be effective if it is not actually requested and cached by altruistic client u. Therefore, we score every content $P_{\mathrm{j}}E_{\mathrm{j}}$ and sort them in the descending of this score.

Second, to optimize $\mathbb{D}_{\mathrm{u}}$, we score the cached content of altruistic client u and the score is defined as $P_{\mathrm{k}}\Delta_{\mathrm{k}}^+$. Δ_{k}^+ is the traffic increased by discarding content k and is obtained from Alg. 3. Why P_{k} needs to be considered is because, in our model described in Sect. 2, altruistic client u can request the content cached in her or his cache space; we can increase P_{comb} by attaching content already cached at client u with larger P_{k} to the combined files while discarding content with smaller P_{k}. Therefore, as in Alg. 4, we sort cached content at client u in the ascending of $P_{\mathrm{k}}\Delta_{\mathrm{k}}^+$.

In Alg. 4, b_{g} is the identification number of the content with the g-th largest $P_{\mathrm{j}}E_{\mathrm{j}}$ while d_{h} is the identification number of a content with the h-th smallest $P_{\mathrm{k}}\Delta_{\mathrm{k}}^+$. x represents the number of content files included in $\mathbb{B}_{\mathrm{u}}$. R_{x} is the integrated score of the combination of $\mathbb{B}_{\mathrm{u}}$ and $\mathbb{D}_{\mathrm{u}}$ when the number of content files included in $\mathbb{B}_{\mathrm{u}}$ is x. Larger R_{x} can reduce more traffic. $R_{\max}$ is the largest R_{x} where $x = X_{\max}$. C is the cache capacity of altruistic client u and x_{c} is how

Algorithm 4 Algorithm for determination of the combined contents

Require: C, $\mathbb{C}$ and $\mathbb{C}_\mathrm{u}$ is known.
sort contents included in $\mathbb{C}$ in descending order of powers of $P_\mathrm{j} \cdot E_\mathrm{j}$
$b_\mathrm{g} \leftarrow$ contents No. whose $P_\mathrm{j} \cdot E_\mathrm{j}$ is the g-th largest.

sort contents included in $\mathbb{C}_u$ in ascending order of powers of $P_\mathrm{k} \cdot \Delta^+_\mathrm{k}$
$d_\mathrm{h} \leftarrow$ contents No. whose $P_\mathrm{k} \cdot \Delta^+_\mathrm{k}$ is the h-th smallest.

$P_\mathrm{comb} \leftarrow 0$
$x \leftarrow 1$
$X_\mathrm{max} \leftarrow x$
$x_\mathrm{c} \leftarrow C - |\mathbb{C}_\mathrm{u}|$
for all i in b_1 and $d_{\max(1,2-x_\mathrm{c})\to(C-x_\mathrm{c})}$ **do**
  $P_\mathrm{comb} \leftarrow P_\mathrm{comb} + P_\mathrm{i}$
end for
if $x_\mathrm{c} <= 0$ **then**
  $E_\mathrm{temp} \leftarrow E_{\mathrm{b}_1} - T\Delta^+_{\mathrm{d}_1} + B_{\mathrm{ub}_1}$
else
  $E_\mathrm{temp} \leftarrow E_{\mathrm{b}_1} + B_{\mathrm{ub}_1}$
end if
$R_\mathrm{x} \leftarrow P_\mathrm{comb} E_\mathrm{temp}$
if $R_\mathrm{x} < 0$ **then**
  $\mathbb{B}_\mathrm{u} \leftarrow \phi$
  $\mathbb{D}_\mathrm{u} \leftarrow \phi$
  return($\mathbb{B}_\mathrm{u}, \mathbb{D}_\mathrm{u}$);
end if
$R_\mathrm{max} \leftarrow R_\mathrm{x}$
for $x = 2$ **to** C **do**
  if $E_{\mathrm{b}_\mathrm{x}} < 0$ **then**
    break
  else
    if $x_c < x$ **then**
      $P_\mathrm{comb} \leftarrow P_\mathrm{comb} + P_{\mathrm{b}_\mathrm{x}} - P_{d(x-x_c)}$
      $E_\mathrm{temp} \leftarrow E_\mathrm{temp} + E_{\mathrm{b}_\mathrm{x}} - T\Delta^+_{\mathrm{d}_\mathrm{x}}$
    else
      $P_\mathrm{comb} \leftarrow P_\mathrm{comb} + P_{\mathrm{b}_\mathrm{x}}$
      $E_\mathrm{temp} \leftarrow E_\mathrm{temp} + E_{\mathrm{b}_\mathrm{x}}$
    end if
    $R_\mathrm{x} \leftarrow P_\mathrm{comb} E_\mathrm{temp}$
    if $R_\mathrm{x} > R_\mathrm{max}$ **then**
      $R_\mathrm{max} \leftarrow R_\mathrm{x}$
      $X_\mathrm{max} \leftarrow x$
    end if
  end if
end for
$\mathbb{B}_\mathrm{u} \leftarrow (b_1, b_2, \ldots, b_{X_\mathrm{max}-1}, b_{X_\mathrm{max}})$
if $x_\mathrm{c} < x$ **then**
  $\mathbb{D}_\mathrm{u} \leftarrow (d_1, d_2, \ldots, d_{\mathrm{x}-\mathrm{x_c}-1}, d_{\mathrm{x}-\mathrm{x_c}})$
else
  $\mathbb{D}_\mathrm{u} \leftarrow \phi$
end if
return($\mathbb{B}_\mathrm{u}, \mathbb{D}_\mathrm{u}$);

many content files client u can additionally cache without discarding any content. E_temp is just temporally used. Readers may notice that, in Alg. 4, B_{ub_1} is eventually canceled. B_{ub_1} is the instantaneous traffic volume increased by peer u when retrieving content b_1. That is why B_{ub_1} is subtracted in Eq. (4) when $j = b_1$. On the other hand, any client requests at least a content file if it gets an opportunity to make a request at a unit time. Therefore, B_{ub_1} is commonly generated and is not considered as additional traffic. That is why B_{ub_1} is added in line 14th or 16th of Alg. 4.

Table 1 Analysis parameters. The parameters are variable.

No. of content files	1000
No. of local domains (n_1)	50
No. of clients in each local domain (n_2)	40
Total no. of clients (N)	2000
Ratio of altruistic clients	10%
Cache capacity at each client (C)	50
Traffic weight B_3, B_2, B_1, B_0	2000, 50, 1, 0

4. Numerical Evaluation

We analyze the upper-bound of the performance of our scheme. The upper-bound of the performance is calculated when we assume the optimality of the combined files is guaranteed during T in Eq. (4); no cache replacement is done during T. In our method, combined files are optimally selected at time t. The optimality of the combined and discarded content files selected by the algorithms in the previous section is effective only for cache state S_t at time t. However, during a certain short period after time t, content files cached in clients are not drastically changed compared with S_t.

We used the parameters listed in Table 1. B_3, B_2, and B_1 indicate traffic impact at the source server, a global domain, and a local domain. B_3, B_2 and B_1 were set corresponding to the numbers of clients managed and handled by the source server, each global domain, and each local domain, respectively [27].

4.1 Analytical Model

We observe how our scheme reduces traffic compared with the case where we do not use our scheme where altruistic clients just request content as non-altruistic clients. In both schemes, it is perfectly known which client is altruistic or not, altruistic clients transfer content files to other clients when their cached content files are requested, and the nearest altruistic client becomes a server for the client requesting a content file. Only the difference between the conventional and proposed schemes is that the proposed scheme provides combined content files to control the request probabilities of altruistic clients and a set of combined content files can be available using only a coupon.

We define the gain of our scheme as:

$$\Psi = \frac{T\left(\zeta\left(S^{\overline{C}}_{\mathrm{t}+1}\right) - \zeta\left(S^{C}_{\mathrm{t}+1}\right)\right) - \eta\left(\mathbb{B}_\mathrm{u}\right)}{T \cdot \zeta\left(S^{\overline{C}}_{\mathrm{t}+1}\right)} \tag{5}$$

where $S^{C}_{\mathrm{t}+1}$ is the state of the cache in the entire network at time $t+1$ when using our scheme, while $S^{\overline{C}}_{\mathrm{t}+1}$ is the one when not using our scheme.

We observe how many content files are combined and from which cached location they are retrieved by an altruistic client: the cache of the altruistic client, the local domain, the global domain, or the source server. We also observe the probability that the content files are retrieved by the altruistic and non-altruistic clients from each location.

4.2 Results

4.2.1 vs. Period T

Let us first analyze the performance of our scheme for different periods T. Figure 5 plots the gain defined in Eq. (5) as a function of T. As shown in this figure, as period T increases, the gain increases. Since T means how long the optimality of content selection of the algorithms in Sect. 3 is sustained, the gain should increase if the optimality can be sustained longer, and this is consistent with our intuition. In reality, even during T, cached content files at clients are replaced and optimality is not guaranteed. Therefore, we observe how largely the cached content files are replaced after period T. Figure 5 plots the correlation of the cached content files at clients in the network at T with the initial cached content files; if no replacement is done during T, the correlation would be 1.0. We investigated two cache replacement algorithms: first-in/first-out (FIFO) and least recently used (LRU). We found that the correlation monotonically decreases as T increases and there is no significant difference between the two replacement algorithms. For the following upper-bound analysis, we set T to be 20,000 because we believe that our approximation would be effective as long as the correlation remains exceeds 90%.

4.2.2 vs. Ratio of Traffic Weight

Figure 6 shows the gain and the number of combined content files versus traffic weight when T = 20,000. The traffic weight (B_3, B_2, B_1, B_0) is as defined in Sect. 2. "Global" and "Source" mean the number of combined content files from the global domain and the source server illustrated in Fig. 2, respectively. In Fig. 6, we can see that, as the traffic weight of the higher layer becomes larger, the gain also increases. This is just simply because the impact of traffic localization increases. On the other hand, although the gain increased from 9.3% to 13.5% by varying traffic weight from (2000, 50, 1, 0) to (10000, 100, 1, 0), the number of combined content files from the source server and the global domains did not change. This is because, as we discussed regarding Eqs. (3) and (4), combining content files increases instantaneous traffic, while our algorithm optimizes the number of combined content files.

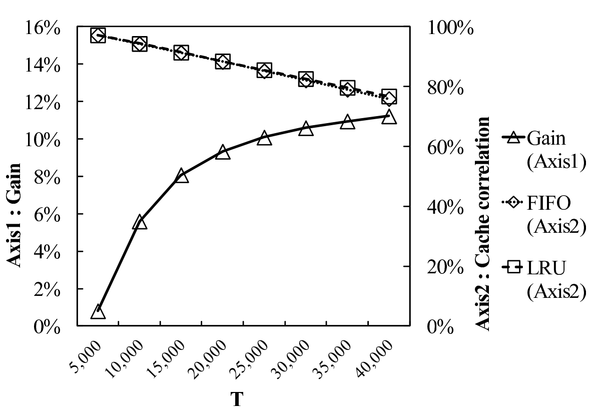

Fig. 5 Gain and cache correlation as a function of the period T. The solid line indicates gain, while the dashed line indicates cache correlation.

4.2.3 vs. Ratio of Altruistic Clients

Figure 7 plots the gain and the number of combined content files as a function of the ratio of altruistic clients when T = 20,000. Table 2 (a) and (b) list the ratio of content sources versus the ratio of altruistic clients. Here, "content source" means where the clients download the content files from. 'SS', 'GD', 'LD' and 'CC' indicate the source server, the global domain, the local domain and their own caches, respectively. In Table 2 (b), altruistic clients request combined contents stochastically. Therefore, we can see that in our scheme SS decreased and traffic was successfully localized. In Fig. 7, we can see that the gain and the total number of combined content files had a peak at the ratio of 10%. This is because when the ratio of altruistic clients exceeds

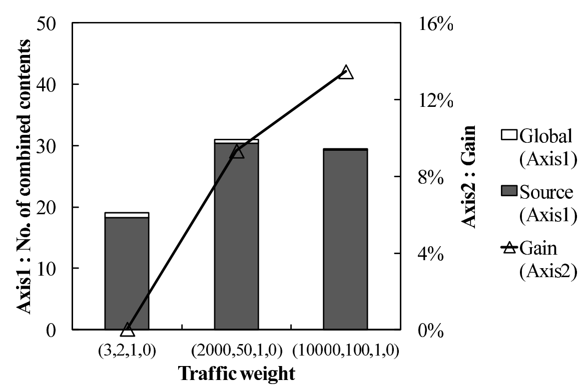

Fig. 6 Number of combined content files and gain as a function of traffic weight. The traffic weight (B_0, B_1, B_2, B_3) was set to (3, 2, 1, 0), (2000, 50, 1, 0) and (10000, 100, 1, 0). The bar graph indicates number of combined content files, while the line graph indicates gain.

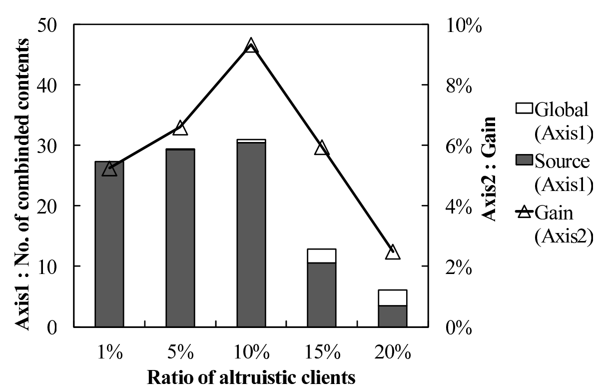

Fig. 7 Gain and number of combined content files as a function of the ratio of altruistic clients when T = 20,000. The bar graph indicates number of combined content files, while the line graph indicates gain.

Table 2 Ratio of content sources vs. ratio of altruistic clients when T = 20,000.

(a) w/o our scheme

Ratio of altruistic clients	SS	GD	LD	CC
1%	15.7%	34.1%	4.1%	46.1%
5%	2.9%	36.2%	14.8%	46.1%
10%	0.6%	31.1%	22.2%	46.1%
15%	0.2%	27.0%	26.8%	46.1%
20%	0.1%	23.8%	30.0%	46.1%

(b) with our scheme

Ratio of altruistic clients	SS	GD	LD	CC
1%	14.8%	35.1%	4.1%	46.1%
5%	2.5%	36.5%	14.8%	46.1%
10%	0.4%	31.3%	22.2%	46.1%
15%	0.1%	27.0%	26.8%	46.1%
20%	0.0%	23.8%	30.1%	46.1%

(c) Content push to altruistic client

Ratio of altruistic clients	SS	GD	LD	CC
1%	13.2%	36.6%	4.0%	46.1%
5%	2.0%	37.1%	14.9%	46.1%
10%	0.1%	31.6%	22.2%	46.1%
15%	0.0%	27.1%	26.8%	46.1%
20%	0.0%	23.8%	30.1%	46.1%

15%, since many contents have found in the cache of the altruistic clients, traffic has already been localized without our scheme. Therefore, the number of combined content files from the source server decreased while that from the global domain increased as the ratio of altruistic clients increased from 15% to 20%. Our scheme cannot further localize traffic, which can be seen in Table 2 (a), (b).

In Table 2 (c), we show the result of the case where we push the set of content files combined for an altruistic client to her or him, which is equivalent to our scheme with $P_{\mathrm{comb}} = 1.0$. This approach may be undesired as we stated in Sect. 1. However, as in Table 2 (c), more traffic localization is observed compared with our scheme in Table 2 (b). This difference is caused by the fact that altruistic clients do not always request combined files; they just request according to the request probabilities as they do for a single content file.

4.2.4 vs. Cache Capacity

Figure 8 plots the gain and the number of combined content files as a function of the ratio of the cache capacity when T = 20,000. Table 3 lists the ratio of the content sources versus cache capacity of each client. In Fig. 8, the gain and the total number of combined content files have a peak when cache capacity is 50. Moreover, the number of combined content files from the source server decreased while that from the global domain increased as the cache capacity increased from 50 to 80. We see a similar trend between Figs. 7 and 8. This is because a larger number of altruistic clients and a larger number of cached content files at each client allow the network to have more content files locally, which is seen in Table 3. However, differently from a small number of altruistic clients, a small number of cached content files limits how many content files can be combined in our scheme. That is why, in Fig. 8, the total number of combined content files decreases, as the cache capacity decreases from 50 to 30.

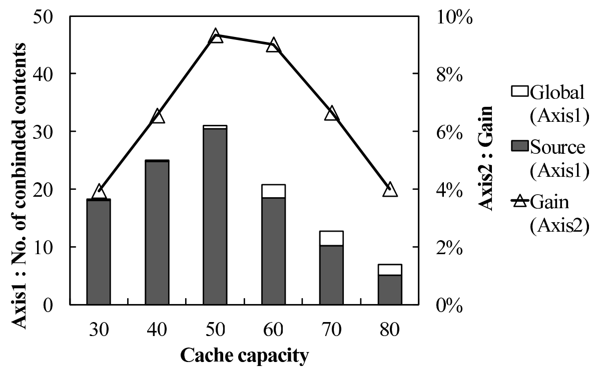

Fig. 8 Gain and number of combined content files as a function of the cache capacity when T = 20,000. The bar graph indicates number of combined content files, while the line graph indicates gain.

Table 3 Ratio of content source vs. cache capacity when T = 20,000.

(a) w/o our scheme

Cache capacity	SS	GD	LD	CC
30	2.6%	37.1%	22.4%	37.9%
40	1.3%	33.9%	22.3%	42.5%
50	0.6%	31.1%	22.2%	46.1%
60	0.3%	28.6%	22.0%	49.1%
70	0.2%	26.3%	21.8%	51.7%
80	0.1%	24.4%	21.7%	53.9%

(b) with our scheme

Cache capacity	SS	GD	LD	CC
30	2.4%	37.3%	22.4%	37.9%
40	1.1%	34.1%	22.3%	42.5%
50	0.4%	31.3%	22.2%	46.1%
60	0.2%	28.7%	22.0%	49.1%
70	0.1%	26.4%	21.9%	51.7%
80	0.0%	24.4%	21.7%	53.9%

4.2.5 vs. the Total Number of Content Files

Figure 9 plots the gain and the number of combined content files as a function of the total number of content files when T = 20,000. Table 4 lists the ratio of content sources versus the total number of content files. The gain has a peak around 900 contents. When the total number of content files ranges from 500 to 900, many contents have been stored by altruistic clients and our scheme can only localize a little bit of traffic, as can be seen in Table 4 (a), (b). Furthermore, when the total number of content files is 1000 to 2000, clients easily request content files not included in the combined content files selected by our scheme. In Fig. 9, the total number of combined content files remains almost constant when the total number of content files is larger than 900; the number of combined content files from the source server increases

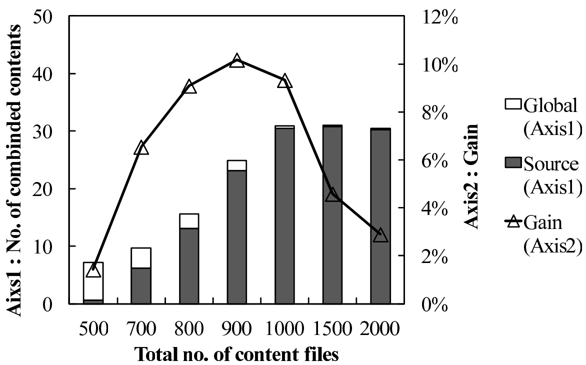

Fig. 9 Gain and number of combined content files as a function of the total number of content files when $T = 20{,}000$. The bar graph indicates the number of combined content files, while the line graph indicates gain.

Table 4 Ratio of content sources vs. no. of contents when $T = 20{,}000$.

(a) w/o our scheme

No. of contents	SS	GD	LD	CC
500	0.0%	23.1%	23.5%	53.4%
700	0.1%	27.3%	22.9%	49.6%
800	0.3%	28.8%	22.6%	48.3%
900	0.4%	30.1%	22.4%	47.1%
1000	0.6%	31.1%	22.2%	46.1%
1500	1.9%	34.2%	21.3%	42.6%
2000	3.4%	35.6%	20.6%	40.4%

(b) with our scheme

No. of contents	SS	GD	LD	CC
500	0.0%	23.1%	23.5%	53.4%
700	0.1%	27.4%	22.9%	49.6%
800	0.2%	28.9%	22.6%	48.3%
900	0.3%	30.2%	22.4%	47.1%
1000	0.4%	31.3%	22.2%	46.1%
1500	1.8%	34.4%	21.3%	42.6%
2000	3.2%	35.8%	20.6%	40.4%

while that from the global domain decreases as the total number of content files increases. As we explained in the previous section, our scheme likely chooses high popularity content files as the combined files. Therefore, this result implies our scheme selects only a certain number of highest popularity content files even when the total number of content files is large.

5. Related Work

5.1 Content Placement

Since the cache capacity of a node is limited, we are not allowed to cache all content at every node. Therefore, we have to consider how to optimize the placement of content in the cache. Content placement in caching networks including CDNs and peer-to-peer (P2P) networks is a classical and well-studied problem [13]–[21]. In CDNs, basically, the content-service providers manage caching networks and control content placement in them. Therefore, the problem in CDNs falls under the policy and algorithm design [13]–[17]. On the other hand, in P2P networks, since no central entity controls the cache placement, distributed approaches have been considered [18]–[21]; each peer decides whether to cache the received content when it obtains the requested content or forwards the content requested by another peer. In peer-assisted CDNs, unlike CDNs and P2P networks, a central entity can attempt to control the cache placement [18], but clients may refuse to cache the directed content files because peer-assisted CDNs owe storage resources to clients. Therefore, our approach does not directly control the cache in the clients but does only induce altruistic clients to cache desired content for traffic localization.

5.2 Conventional Incentive Mechanisms in Peer-Assisted Services

Our motivation to introduce an incentive mechanism is quite different from most previous studies on P2P or peer-assisted networks. Conventionally, we need to introduce it because contributions by peers are essential in P2P and peer-assisted networks; without such contributions, the services would not provide any benefits to the clients. However, as reported in [8], most peers are free riders who do not contribute their resources to the networks. Thus, motivating them has been the purpose of the previous work on incentive mechanisms [22]–[26]. However, unlike previous efforts, our purpose is simply to induce altruistic clients to request specific content, that will likely be requested on the local network and will likely reduce traffic. The form of the incentive is another factor in which we are interested. Some systems give incentives as service quality [22], [24], [25], and others provide monetary incentives [23], [26]. However, the effectiveness of these approaches is mathematically unclear; it is unclear how much a certain level of increased service quality or money increases the probability that a free rider will contribute to the network. However, incentive by content combination is straight forward. First, the request probability of the combined content files is equal to or larger than the sum of the request probabilities of each content. Second, as mentioned in Sect. 2, since clients are charged at a fixed rate to obtain coupons at every period and the combined content is just an electric copy of the original file, unlike a monetary incentive, we can ignore the source of the incentive reward. In other words, we do not need to consider how much economically we gained or lost by giving incentives to clients.

6. Conclusion

This paper proposed a novel traffic engineering scheme for peer-assisted CDN models that optimizes content files at altruistic clients to optimally localize traffic. Its key idea is to control the behavior of clients by using content-oriented incentive mechanism which combines content files while keeping the price equal to the single-content price to induce altruistic clients to request files desired to be cached. We reveal the trade-off between the increase in traffic resulting from content combination and the reduction in traffic resulting from traffic localization. Considering the trade-off, we formulated a problem for the optimal selection of combined content files and derive a solution for the optimal selection of combined content files for altruistic clients, and describe it as algorithms for solving the problem. Our numerical analysis observed the upper-bound of the performance traffic localization by content combination while taking into account the trade-off, and confirmed that our scheme effectively localizes traffic and reduces overall traffic.

Our future work will include an evaluation of our scheme in a realistic simulation and an actual implementation of our scheme.

References

[1] G. Peng, "CDN: Content distribution network," Dept. Computer Science, State Univ. of New York, New York, Tech. Rep. TR-125, 2003.
[2] G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," Commum. ACM, vol.49, no.1, pp.101–106, 2006.
[3] "Akamai," http://www.akamai.com/index.html?intl=1
[4] T. Mori, N. Kamiyama, S. Harada, H. Hasegawa, and R. Kawahara, "Improving deployability of peer-assisted CDN platform with incentive," Proc. IEEE GLOBECOM'09, pp.1–7, Honolulu, Nov. 2009.
[5] D. Xu, S. Kulkarni, C. Rosenberg, and H. Chai, "Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution," Mutimedia Syst., vol.11, no.4, pp.383–399, March 2006.
[6] C. Huang, A. Wang, J. Li, and K.W. Ross, "Understanding hybrid CDN-P2P: Why limelight needs its own red swoosh," Proc. 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'08), pp.75–80, Braunschweig, Germany, May 2008.
[7] "BitTorrent DNA," http://www.bittorrent.com/dna
[8] E. Adar and B. Huberman, "Free riding on gnutella," First Monday, vol.5, no.10, Oct. 2000.
[9] D. Hughes, G. Coulson, and J. Walkerdine, "Freeriding on gnutella revisited: The bell tolls?," IEEE DS Online, June 2005.
[10] T. Bocek, M. Shann, D. Hausheer, and B. Stiller, "Game theoretical analysis of incentives for large-scale, fully decentralized collaboration networks," IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008), pp.1–8, Miami, April 2008.
[11] R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, A. Guerrero, and R. Rejaie, "Is content publishing in Bittorrent altruistic or prifit-driven?," ACM CoNEXT, 2010.
[12] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," Proc. INFOCOM'99, vol.1, pp.126–134, New York, March 2002.
[13] X. Tang and S.T. Chanson, "Coordinated en-route web caching," IEEE Trans. Comput., vol.51, no.6, pp.595–607, June 2002.
[14] A. Nakaniwa, H. Ebara, and H. Okada, "File allocation designs for distributed multimedia information networks," Proc. IEEE GLOBECOM'98, vol.2, pp.740–745, Sydney, Nov. 1998.
[15] F.L. Presti, N. Bartolini, and C. Petrioli, "Dynamic replica placement and user request redirection in content delivery networks," Proc. IEEE ICC, vol.3, pp.1495–1501, Seoul, May 2005.
[16] A. Jiang and J. Bruck, "Optimal content placement for en-route web caching," Proc. 2nd International Symposium on Network Computing and Applications, pp.9–16, Pasedena, April 2003.
[17] Y. Chen, L. Qiu, W. Chen, L. Nguyen, and R.H. Katz, "Efficient and adaptive web replication using content clustering," IEEE J. Sel. Areas Commun., vol.21, no.6, pp.979–994, Aug. 2003.
[18] N. Kamiyama, R. Kawahara, T. Mori, and H. Hasegawa, "Multicast pre-distribution in VoD service," Proc. IEEE CQR, Naples, May 2011.
[19] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," Proc. ACM SIGCOMM'02, Pittsburgh, Aug. 2002.
[20] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer system," Proc. IEEE INFOCOM'03, vol.3, pp.2166–2176, Pittsburgh, April 2003.
[21] J. Kangasharju, K.W. Ross, and D.A. Turner, "Optimizing file availability in peer-to-peer content distribution," Proc. IEEE INFOCOM'07, pp.1973–1981, Anchorage, May 2007.
[22] M. Yamada, K. Sato, R. Shinkuma, and T. Takahashi, "Incentive service differentiation for p2p content sharing by wireless users," IEICE Trans. Commun., vol.E90-B, no.12, pp.3561–3571, Dec. 2007.
[23] K. Sato, R. Hashimoto, M. Yoshino, R. Shinkuma, and T. Takahashi, "Incentive mechanism for P2P content sharing over heterogeneous access networks," IEICE Trans. Commun., vol.E91-B, no.12, pp.3821–3830, Dec. 2008.
[24] S. Jun and M. Ahamad, "Incentives in Bittorrent induce free riding," Proc. 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems, pp.116–121, Philadelpha, Aug. 2005.
[25] R. Cheng and J. Vassileva, "User motivation and persuasion strategy for peer-to-peer communities," Proc. 38th Annual Hawaii International Conference on System Science (HICSS'05), pp.193–202, Hawaii, Jan. 2005.
[26] C. Wang, H. Wang, Y. Lin, and S. Chen, "A currency-based p2p incentive mechanism friendly with ISP," Proc. Computer Design and Applications (ICCDA), vol.5, pp.403–407, Qinhuangdao, June 2010.
[27] K. Cho, K. Fukuda, H. Esaki, and A. Kato, "The impact and implications of the growth in residentaial user-to-user traffic," Proc. ACM SIGCOMM'06, pp.207–218, Pisa, Sept. 2006.

Naoya Maki received the B.E. degree in Electrical and Electronic Engineering from Kyoto University, Kyoto, Japan, in 2011. He is a master course student of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University. His current research interest is in traffic engineering for peer-assisted content delivery networks.

Takayuki Nishio received the B.E. degree in Electrical and Electronic Engineering from Kyoto University in 2010, and the M.I. degree in Communications and Computer Engineering, Graduate School of Informatics from Kyoto University, Kyoto, Japan, in 2012. He is currently working toward the Ph.D. degree in Communications and Computer Engineering, Graduate School of Informatics, Kyoto University. His current research interests include QoS and resource management in wireless networks.

Ryoichi Shinkuma received the B.E., M.E., and Ph.D. degrees in Communications Engineering from Osaka University, Osaka, Japan, in 2000, 2001, and 2003, respectively. In 2003, he joined Communications and Computer Engineering, Graduate School of Informatics, Kyoto University as an Associate Professor. He was a Visiting Scholar at Wireless Information Network Laboratory (WINLAB), Rutgers, the State University of New Jersey, USA, from 2008 Fall to 2009 Fall. His research include network design and control criteria, particularly inspired by economic and social aspects. He received the Young Researchers' Award from IEICE in 2006 and the Young Scientist Award from Ericsson Japan in 2007, respectively. He is a member of IEEE.

Tatsuya Mori is currently a senior researcher at NTT Network Technology Laboratories, Tokyo, Japan. He received B.E. and M.E. degrees in applied physics, and Ph.D. degree in information science from the Waseda University, Tokyo, Japan, in 1997, 1999 and 2005, respectively. Since joining NTT corporation in 1999, he has been engaged in the research of management of large-scale networked systems and network security. From March 2007 to March 2008, he was a visiting researcher at the University of Wisconsin-Madison. He received Telecom System Technology Award from TAF in 2010 and Best Paper Awards from IEICE and IEEE/ACM COMSNETS in 2009 and 2010, respectively. Dr. Mori is a member of ACM and IEEE.

Noriaki Kamiyama received his M.E. and Ph.D. degrees in communications engineering from Osaka University in 1994 and 1996, respectively. From 1996 to 1997, he was with the University of Southern California as a visiting researcher. He joined Multimedia Networks Laboratories, in 1997. Since then, he has been engaged in research concerning pricing methods for IP networks, content distribution systems, optical networks, and IP traffic measurement. He is currently working on traffic research in NTT Network Technology Laboratories. He is a member of IEEE and the Operations Research Society of Japan. He received Best Paper Awards form IEICE in 2003 and 2009.

Ryoichi Kawahara received a B.E. in automatic control, an M.E. in automatic control, and a Ph.D. in telecommunication engineering from Waseda University, Tokyo, Japan, in 1990, 1992, and 2001, respectively. Since joining NTT in 1992, he has been engaged in research on traffic control for telecommunication networks. He is currently working on teletraffic issues in IP networks in NTT Network Technology Laboratories. He is a member of IEEE and the Operations Research Society of Japan.

Tatsuro Takahashi received the B.E., M.E. in Electrical Engineering from Kyoto University, Kyoto, Japan, in 1973 and 1975 respectively, and Dr. of Engineering in Information Science from Kyoto University in 1975 to 2000, making R&D on high speed networks and switching systems for circuit switching, packet switching, frame relaying, and ATM. Since July 1, 2000, he is a Professor, Communications and Computer Engineering, Graduate School of Informatics, Kyoto University. His current research interests include high-speed networking, photonic networks and mobile networks. Prof. Takahashi received the Achievement Award from IEICE in 1996, the Minister of Science and Technology Award in 1998, and the Distinguished Achievement of Contributions Award from IEICE in 2011. He was a Vice President of the ATM Forum from 1996 to 1997, and the Chairman of the Network Systems (NS) Technical Group in the Communications Society of IEICE from 2001 to 2002. Prof. Takahashi is an IEEE Fellow.