Data Broadcast on a Multi-System Heterogeneous Overlayed Wireless Network*

JIUN-LONG HUANG AND JUI-NAN LIN
Department of Computer Science
National Chiao Tung University
Hsinchu, 300 Taiwan
E-mail: {jlhuang; jnlin}@cs.nctu.edu.tw

A multi-system heterogeneous overlayed wireless network with multiple wireless access technologies is deemed a key part of 4*G* networks. In addition, data broadcast is a promising technique to improve the bandwidth utilization and to conserve the power consumption in a mobile computing environment. However, most of the prior studies in data broadcast only deal with issues in a single-system wireless network (*i.e.*, in a network with one or multiple broadcast channel(s)), and therefore, these prior approaches cannot be directly used in a multi-system heterogeneous overlayed wireless network. In view of this, we propose in this paper a two-phase algorithm, named algorithm Layered-Cutting, to address the problem of broadcast program generation in a multi-system heterogeneous overlayed wireless network. Specifically, in inter-network data allocation phase, algorithm Layered-Cutting allocates a set of data items to each subnetwork. Then, in intra-network data allocation phase, algorithm Layered-Cutting generates one broadcast program for each subnetwork according to the number of channels in the subnetwork and the properties (including data access probabilities and object sizes) of the data items allocated to the subnetwork. To evaluate the performance of algorithm Layered-Cutting, several experiments are conducted. The experimental results show that algorithm Layered-Cutting is able to efficiently generate broadcast programs of high quality for a multi-system heterogeneous overlayed wireless network.

Keywords: data broadcast, heterogeneous overlayed network, mobile computing, mobile data management, ubiquitous computing

1. INTRODUCTION

A multi-system heterogeneous wireless network with multiple wireless access technologies is deemed a key part of 4*G* networks [1, 2]. A 4*G* network can be conceptually visualized as a collection of multiple independent access subnetworks. This vision leverages the relative merits of multiple cellular access systems, with significant heterogeneity in their individual characteristics such as coverage area, transmission range and channel bandwidth. By using multiple, physical or software-defined radio interfaces, mobile devices are able to switch between these wireless access technologies to obtain better service quality. In ubiquitous computing environments, the service providers should take advantage of the relative merits of different subnetworks to provide high quality network access at anytime in anywhere.

Received June 23, 2006; revised October 16, 2006; accepted January 3, 2007.
Communicated by Chung-Sheng Li.
* The preliminary version has been published in the 7th IEEE International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06) 2006.

In order to provide power conserving and high scalable services in a mobile environment, a data delivery architecture in which a server continuously and repeatedly broadcasts data to a client community through a single broadcast channel was proposed in [3, 4]. Related research issues about broadcast-based information systems have attracted a considerable amount of studies, including broadcast program generation [3, 5], on-demand broadcast [6-8], data indexing [9, 10], location-dependent data broadcasting [11, 12], dynamic data and channel allocation [13, 14], and dependent data broadcasting [15-18].

Unfortunately, most of the prior studies in data broadcast only deal with data indexing, broadcast program generation or other issues in a *single-system wireless network* (*i.e.*, in *one* network with one or multiple broadcast channel(s)). Therefore, the approaches proposed by these studies cannot be directly used in *a multi-system heterogeneous overlayed wireless network*. We argue that with the development of 4*G* networks, designing a proper scheme to employ data broadcast on multi-system networks will become an important issue in the development of mobile information systems.

In view of this, we propose a two-phase algorithm, named algorithm Layered-Cutting, to address the problem of broadcast program generation in a multi-system heterogeneous overlayed wireless network. Specifically, algorithm Layered-Cutting consists of two phases: inter-network data allocation and intra-network data allocation, and cooperates with a designated traditional broadcast program generation algorithm for a single-system network. For better readability, the employed traditional broadcast program generation algorithm for a single-system network is referred to as algorithm BPG-Single (standing for Broadcast Program Generation for Single-system networks) in the rest of this paper. In inter-network data allocation phase, algorithm Layered-Cutting allocates a set of data items to each subnetwork. Since broadcast program generation algorithms are usually of high complexity, for better scalability, the times of executing algorithm BPG-Single should be minimized. To achieve this, algorithm Layered-Cutting employs an overall average access time estimation method to estimate the quality of different data allocation settings, and determines a data allocation setting with smaller *estimated* overall access time as the result of inter-network data allocation phase. After determining a proper data allocation setting in inter-network data allocation phase, algorithm Layered-Cutting steps into intra-network data allocation phase to generate one broadcast program for each subnetwork according to the number of channels in the subnetwork and the properties (including data access probabilities and object sizes) of the data items allocated to the subnetwork. To evaluate the performance of algorithm Layered-Cutting, several experiments are conducted. The experimental results show that algorithm Layered-Cutting is able to efficiently generate broadcast programs of high quality for a multi-system heterogeneous overlayed wireless network. To the best of our knowledge, there is no prior work considering data broadcast on heterogeneous overlayed networks. This characteristic distinguishes this paper from others.

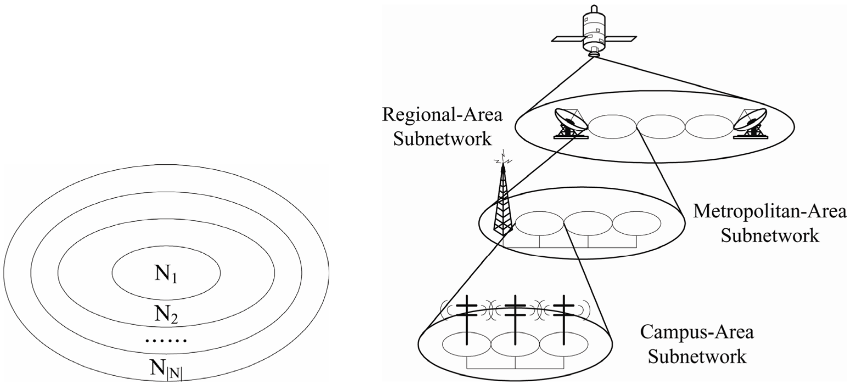
The rest of this paper is organized as follows. First, problem description and formulation are given in section 2. Then, the details of algorithm Layered-Cutting are shown in section 3. Section 4 shows the experimental results, and finally, section 5 concludes this paper.

2. PRELIMINARIES

2.1 System Model

Consider a multi-system heterogeneous overlayed network $N = \{N_1, N_2, \ldots, N_{|N|}\}$ consisting of $|N|$ subnetworks, and suppose that these subnetworks are ordered by the sizes of their coverage areas in ascending order. To facilitate the following discussion, we make the following two assumptions.

1. The service area of subnetwork N_i is totally covered by that of subnetwork N_j if $j > i$.
2. When being able to connect to subnetwork N_i and subnetwork N_j simultaneously, users prefer using subnetwork N_i than using subnetwork N_j if $j > i$.



(a) General system model. (b) Example.
Fig. 1. Multi-system heterogeneous wireless networks.

With the above assumptions, a general system model of multi-system heterogeneous overlayed networks is shown in Fig. 1 (a), while an example is shown in Fig. 1 (b). These two assumptions hold in many cases. As mentioned in [19], wireless networks with larger coverage areas are usually of higher connection fee and of lower bandwidth than those with smaller coverage areas. For example, the service area of a GPRS network is larger than the service area of a Wi-Fi network, and the service area of a Wi-Fi network is usually totally covered by a GPRS network. In addition, when being able to connect to these two networks, users usually prefer using the Wi-Fi network rather than using the GPRS network since Wi-Fi networks are cheaper and of higher bandwidth than GRPS networks. Interested reader can refer to [19] for the more descriptions of multi-system heterogeneous overlayed networks.

2.2 Problem Description and Formulation

Suppose that in subnetwork N_i, the service provider allocates C_i channels to provide the data broadcast service and the bandwidth of each channel is B_i. Suppose that the database D contains $|D|$ data items, $D_1, D_2, \ldots, D_{|D|}$. Also let the size of D_i be s_i and let the

Table 1. Descriptions of used symbols.

Symbol	Description
N_i	the ith subnetwork
A_i	size of the service area of the ith subnetwork
C_i	number of broadcast channels in subnetwork N_i
B_i	Bandwidth of one broadcast channel in subnetwork N_i
D_i	the ith data item
s_i	size of D_i
l_i	time to broadcast D_i
p_i	access probability of D_i
Cut_i	the ith cutting point
BPG-Single	the selected broadcast program generation for single-system networks

access probability of data item D_i be p_i. For better readability, the descriptions of symbols used are listed in Table 1.

Two metrics, access time and tuning time, are introduced in [4] to evaluate the performance of broadcast programs. The access time is the time elapsed from the moment a client issues a request to the moment all the relevant data are read. The tuning time is the amount of time spent by the client listening on the broadcast channels, which is a measurement of the power consumption. In this paper, we take the access time as the measurement of the performance of broadcast programs. Note that in broadcast environments, "a user issues a request" does not mean that the mobile device has to explicitly issue a data request to the server. In fact, it means that the user issues a data request to the "mobile device," and the mobile device will tune to the broadcast channel, wait for the appearance of the required data item and retrieve the required data item from the broadcast channel.

As mentioned in [20], in a *single-system network* with *one* broadcast channel of bandwidth B, to minimize the overall average access time, instances of each item have to be equally spaced, and the broadcast frequency of data item D_i should be proportional to $\sqrt{\frac{p_i}{l_i}}$, where l_i is defined as the time to broadcast D_i in the broadcast channel. That is, l_i is equal to $\frac{s_i}{B}$ where B is the bandwidth of the broadcast channel. This result is also called *square-root rule*. When square-root rule is satisfied, the lower bound of overall average access time is

$$t_{Opt.} = \frac{1}{2}\left(\sum_{i=1}^{|D|}\sqrt{p_i \times l_i}\right)^2. \tag{1}$$

On the other hand, if the network contains multiple channels, say *ChannelNo* broadcast channels, the lower bound of overall average access time will be

$$t_{Opt.}^{Multi} = \frac{t_{Opt.}}{ChannelNo}. \tag{2}$$

In addition to derive the above theoretical results, the authors in [20] propose a probabilistic broadcast program generation algorithm to generate broadcast programs in a single-channel environment. They also extend the proposed algorithm to a multi-channel environment. For ease of presentation, we assume that all data items have been reordered according to their $\sqrt{\frac{p_i}{l_i}}$ values in descending order in the rest of this paper.

We now consider the cases in a multi-system heterogeneous overlayed wireless network. We first observe the case that data items $D_1, D_2, \ldots, D_{|D|}$ are broadcast in a multi-system heterogeneous overlayed wireless network consisting of two subnetworks, N_1 and N_2. Since N_2 is of the largest service area, all data items have to be broadcast in N_2 in order to provide the highest data availability. On the other hand, to minimize the overall average access time, N_1 will only broadcast some data items with high broadcast frequentcies (*i.e.*, high $\sqrt{\frac{p_i}{l_i}}$ values). Therefore, we have to determine a cutting point Cut_1 so that data items from D_1 to D_{Cut_1} are broadcast in N_1 and data items from D_1 to $D_{|D|}$ (*i.e.*, all data items) are broadcast in subnetwork N_2. Here we say that data items from D_1 to D_{Cut_1} are *allocated* to subnetwork N_1 and data items from D_1 to $D_{|D|}$ are *allocated* to subnetwork N_2. Since access time is taken as the performance metric, we should determine a proper value of Cut_1 to minimize overall average access time.

We then extend the above observation to a more general case with $|N|$ subnetworks. When we have to broadcast data items $D_1, D_2, \ldots, D_{|D|}$ in a multi-system heterogeneous wireless network consisting of $|N|$ subnetworks, $N_1, N_2, \ldots, N_{|N|}$, we shall first determine the values of $|N| - 1$ cutting points, $|Cut_1|, |Cut_2|, \ldots, |Cut_{|N|-1}|$, where $Cut_i \leq Cut_j$ when $i < j$. Then, for $i = 1, 2, \ldots, |N| - 1$, data items from D_1 to D_{Cut_i} are allocated to subnetwork N_i. All data items are allocated to subnetwork $N_{|N|}$. Hence, we have the following definition.

Definition 1 A *cutting configuration* is defined as a setting of the values of Cut_1, $Cut_2, \ldots, Cut_{|N|-1}$ so that (1) $Cut_i \leq Cut_j$ if $i \leq j$ and (2) $1 \leq Cut_i \leq |D|$ for all i.

Since overall average access time is taken as the performance metric in this paper, the determination of the values of these cutting points has to be under the goal of minimizing overall average access time. This procedure is called *inter-network data allocation*, and the flowchart of inter-network data allocation is shown in Fig. 2.

Each subnetwork is assigned some data items after inter-network data allocation. Then, for each subnetwork, we should determine how to broadcast the assigned data items by all broadcast channels of this subnetwork. That is, to generate a broadcast program for each subnetwork. Such procedure is called *intra-network data allocation*. Fortunately, the problem of intra-network data allocation is equivalent to the problem of broadcast program generation in a single-system network with multiple broadcast channels which has been widely studied in many prior studies [5, 21, 22]. Hence, we will focus on inter-network data allocation in the rest of this paper and employ one prior broadcast program generation algorithm in a single-system network with multiple broadcast channels to deal with intra-network data allocation for each subnetwork.

As a consequence, the problem of broadcast program generation on heterogeneous overlayed wireless networks can be formulated as follows.

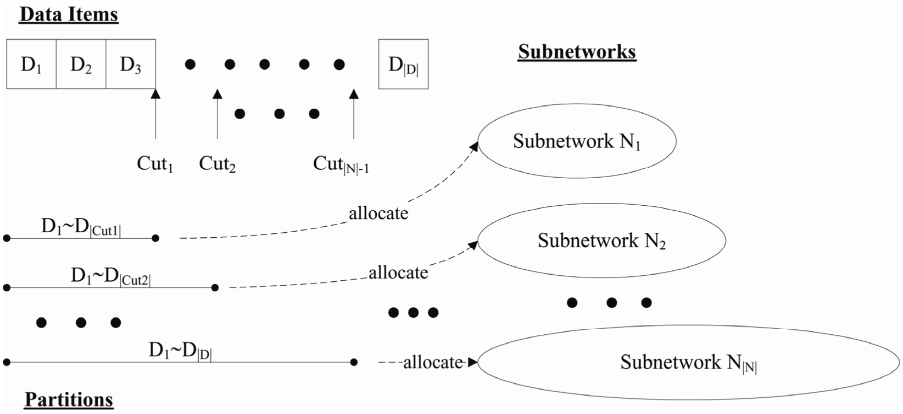


Fig. 2. Flowchart of inter-network data allocation.

Definition 2 Given a multi-system heterogeneous wireless network $N = \{N_1, N_2, \ldots, N_{|N|}\}$, the number of allocated channels C_i in each subnetwork N_i, the number of data items, and the access probabilities and sizes of all data items, for each subnetwork N_i, we shall determine:

1. which data items will be broadcast in subnetwork N_i (*i.e.*, a proper cutting configuration), and
2. how these data items are broadcast in subnetwork N_i (*i.e.*, one proper broadcast program for each subnetwork).

2.3 A Brute Force Algorithm for Optimal Solutions

Since the process in intra-network data allocation is equivalent to generating broadcast programs in a single-system network, we first select a broadcast program generation algorithm for a single-system network, and such algorithm is referred to as algorithm BPG-Single for ease of presentation. Note that algorithm BPG-Single can be any broadcast program generation algorithm (*e.g.*, algorithm VF^K [5]) as long as it can generate broadcast programs for multiple broadcast channels in a single-system network.

We here design a brute force algorithm, named algorithm Brute-Force, to obtain optimal solutions by evaluating all possible cutting configurations and selecting the best cutting configuration and the corresponding broadcast programs for all subnetworks (*i.e.*, with the smallest overall average access time) as the result. When evaluating one cutting configuration, we should generate one broadcast program for each subnetwork by executing algorithm BPG-Single once so that the average access time of the subnetwork is minimized. Then, the overall average access time of the whole multi-system network is determined as the *weighted summation* of the average access time of each subnetwork. Suppose that the data access probabilities of all data items observed by subnetwork N_j are $p_1^j, p_2^j, \ldots, p_{|D|}^j$.[1] The weight of a subnetwork is defined as below.

Definition 3 The weight of subnetwork N_j is defined as the probability that a data

[1] Please refer to Appendix for the method to determine the values of p_i^j.

request is served by subnetwork N_j. Therefore, the weight of subnetwork N_j is equal to $\sum_{i=1}^{|D|} p_i^j$.

After all cutting configurations have been evaluated, the best cutting configuration (*i.e.*, with the smallest overall average access time) and the corresponding broadcast programs for all subnetworks are selected as the result of the algorithm. The algorithmic form of algorithm Brute-Force is as follows.

Algorithm Brute-Force

```
Mark all possible configurations as UNEVALUATED
BEST ← NULL     /* BEST is the configuration of smallest overall average access time
                   up-to-now */
while there are some unevaluated cutting configurations do
    Pick one unevaluated cutting configuration, say CURRENT
    for (i = 1 to |N|) do
    Execute algorithm BPG-Single on subnetwork N_i with cutting configuration CURRENT
    end for
    Calculate the overall average access time of CURRENT
    if (the overall average access time of CURRENT is smaller than that of BEST) then
        BEST ← CURRENT
    end if
    Mark CURRENT as EVALUATED
end while
return BEST
```

Since evaluating all possible cutting configurations, algorithm Brute-Force will evaluate $O(|D|^{|N|-1})$ cutting configurations. In each evaluation, algorithm BPG-Single is executed $|N|$ times to generate broadcast programs for all subnetworks. Let $O(BPG_{Single})$ be the time complexity of algorithm BPG-Single. The time complexity of evaluating one cutting configuration is $|N| \times O(BPG_{Single})$. Therefore, the time complexity of algorithm Brute-Force is $O(|D|^{|N|-1}) \times |N| \times O(BPG_{Single}) = O(|D|^{|N|} \times |N|) \times O(BPG_{Single})$.

3. BROADCAST PROGRAM GENERATION ON A MULTI-SYSTEM HETEROGENEOUS OVERLAYED WIRELESS NETWORK

Although being able to obtain the optimal cutting configuration and the corresponding broadcast programs for all subnetworks, the time complexity of algorithm Brute-Force is quite high. This result makes algorithm Brute-Force not suitable for practical use, and hence, motivates us to design a more efficient algorithm, named algorithm Layered-Cutting in this section to determine a suboptimal cutting configuration and the corresponding broadcast programs for all subnetworks.

3.1 Overview of Algorithm Layered-Cutting

Obviously, the reasons that the time complexity of algorithm Brute-Force is so high are as follows.

1. Algorithm Brute-Force entangles the inter-network and intra-network data allocation together.
   Since being executed several times when a cutting configuration is evaluated, algorithm BPG-Single will be executed many times within one execution of algorithm Brute-Force. Owing to the fact that broadcast program generation algorithms in a single-system network (*e.g.*, algorithm BPG-Single) are usually of high time complexity, the time complexity of algorithm Brute-Force is inherently high.
2. Algorithm Brute-Force searches all possible cutting configurations, and therefore, does not scale well.

To address the above two problems, the design rationales of algorithm Layered-Cutting are as follows.

1. Reduce the number of times of executing algorithm BPG-Single. That is, to reduce the number of times of executions of "the employed broadcast program generation algorithm in a single-system network."
2. Reduce the size of search space by searching only the configurations of high probabilities to be the optimal configuration instead of searching all possible cutting configurations.

Basically, algorithm Layered-Cutting is a two-phase algorithm consisting of inter-network data allocation phase and intra-network data allocation phase. The objective of inter-network data allocation phase is to determine a proper cutting configuration so that the overall average access time of the whole multi-system network is minimized. Then, in intra-network data allocation phase, algorithm Layered-Cutting will generate one broadcast program for each subnetwork according to the resultant cutting configuration. Different from algorithm Brute-Force, algorithm Layered-Cutting does not execute algorithm BPG-Single in inter-network data allocation phase, since the goal of inter-network data allocation phase is only to generate a proper cutting configuration. In intra-network data allocation phase, algorithm Layered-Cutting will execute algorithm BPG-Single only $|N|$ times to generate one broadcast program for each subnetwork. In addition, instead of evaluating all possible cutting configuration (algorithm Brute-Force does it), algorithm Layered-Cutting only evaluates some cutting configurations with high probability to be optimal. With the above two characteristics, algorithm Layered-Cutting is able to obtain suboptimal cutting configurations efficiently.

3.2 Phase One: Inter-Network Data Allocation Phase

According to rationale one, we should reduce the number of times of executing algorithm BPG-Single. Since the objective of inter-network data allocation phase is only to determine a proper cutting configuration to minimize overall average access time, knowing the broadcast programs of all subnetworks is not necessary in inter-network data allocation phase. Therefore, when evaluating a cutting configuration, we use Eqs. (1) and (2) to obtain the lower bound of overall average access time of each subnetwork and take the *weighted* summation of these lower bounds as the lower bound of the whole multi-system network. Similar to algorithm Brute-Force, the weight of a subnetwork is the probability that a data request is served by the subnetwork. By employing Eqs. (1) and (2), algorithm Layered-Cutting can use the lower bounds of the overall average access time of cutting configurations as the *estimated* overall average access time without executing algorithm BPG-Single.

Basically, algorithm Layered-Cutting is an iterative algorithm which merges several subnetworks and determines the value of one cutting point in each iteration. The procedure to merge subnetworks is shown in section 3.2.1. Then, the iterative procedure of algorithm Layered-Cutting is described in section 3.2.2 while the procedure to determine the values of cutting points is given in section 3.2.3.

3.2.1 Merging subnetworks

To facilitate the design of algorithm Layered-Cutting, we first consider the effect of merging some subnetworks into a logical subnetwork. Suppose that the data access probabilities of all data items observed by the combination of subnetworks $N_1, N_2, \ldots, N_j$ are $p_1^{1\sim j}, p_2^{1\sim j}, \ldots, p_{|D|}^{1\sim j}$.[2] We then merge the combination of $N_1, N_2, \ldots, N_j$ into a *logical* single-system subnetwork, denoted as $N_{1\sim j}$, with service area of size A_j and one logical broadcast channel of bandwidth $B_{1\sim j}$, where $B_{1\sim j}$ is determined as follows. Consider a subnetwork N_i, $1 \le i \le j$, which has C_i broadcast channels and each is of bandwidth B_i. The weight of logical subnetwork $N_{1\sim j}$ is defined as below.

Definition 4 The weight of logical subnetwork $N_{1\sim k}$ is defined as the probability that a data request is served by one of subnetworks $N_1, N_2, \ldots, N_{|N|}$. Therefore, the weight of logical subnetwork $N_{1\sim k}$ is equal to $\sum_{i=1}^{|D|} p_i^{1\sim k}$.

In addition, the aggregate bandwidth of subnetwork N_i is $C_i \times B_i$. Since the sizes of service areas of subnetwork N_i and logical subnetwork $N_{1\sim j}$ are A_i and A_j, respectively, the contribution of subnetwork N_i on bandwidth of logical subnetwork $N_{1\sim j}$ can be estimated by uniformly spreading the bandwidth from service area with size A_i to service area with size A_j. Therefore, $B_{1\sim j}$ is defined as the summation of the contributions of subnetwork N_1, $N_2, \ldots, N_j$. As a result, $B_{1\sim j}$ can be formulated as

$$\sum_{i=1}^{j} \frac{A_i}{A_j} \times C_i \times B_i .$$

Therefore, the lower bound of the overall average access time of subnetwork $N_{1\sim j}$ can be obtained by Eq. (1) with data access probabilities $p_1^{1\sim j}, p_2^{1\sim j}, \ldots, p_{|D|}^{1\sim j}$.[2] Finally, the lower bound of overall average access time of the combination of subnetworks N_1, $N_2, \ldots, N_j$ can be approximated by the lower bound of the overall average access time of subnetwork $N_{1\sim j}$.

3.2.2 Layered cutting

The objective of inter-network data allocation phase is to determine a proper cutting

[2] Please refer to Appendix for the method to determine the values of $p_i^{1\sim j}$ details.

configuration (*i.e.*, determine the values of Cut_1, Cut_2, …, $Cut_{|N|-1}$) to minimize overall average access time of the whole multi-system network.

Basically, inter-network data allocation phase of algorithm Layered-Cutting is an iterative algorithm and determines the value of $Cut_{|N|-j}$ in the jth iteration. In the first iteration, we have a multi-system network with $|N|$ subnetworks. Since subnetwork $N_{|N|}$ is of the largest service area, all data requests will be served by the combination of subnetworks N_1, N_2, …, $N_{|N|}$. Hence, we have $p_i^{1\sim|N|} = p_i$ for each data item D_i. We then merge subnetworks N_1, N_2, …, $N_{|N|-1}$ into logical subnetwork $N_{1\sim|N|-1}$ with service area of size $A_{|N|-1}$ and a broadcast channel of bandwidth $B_{1\sim|N|-1}$. As a result, determining the value of $Cut_{|N|-1}$ in a multi-system network with $|N|$ subnetworks is transformed into determining the value of $Cut_{|N|-1}$ in a multi-system network with *two* networks (*i.e.*, logical subnetwork $N_{1\sim(j-1)}$ and subnetwork N_j). We then design a heuristic, named procedure Test-and-Prune, to determine the value of the cutting point between logical subnetwork $N_{1\sim(j-1)}$ and subnetwork N_j. For ease of presentation, we defer the description of procedure Test-and-Prune to section 3.2.3 and assume that value of $Cut_{|N|-1}$ can be obtained right now. We then determine the access probabilities observed by logical subnetwork $N_{1\sim|N|-1}$ (i.e., $p_i^{1\sim|N|-1}$) and by subnetwork $N_{|N|}$ (*i.e.*, $p_i^{|N|}$). After $p_i^{1\sim|N|-1}$ and $p_i^{|N|}$ have been calculated, algorithm Layered-Cutting finishes the first iteration and starts the second iteration.

In essence, the process of the jth iteration is similar to that of the first iteration. In the jth iteration, only subnetworks N_1, N_2, …, $N_{|N|-j+1}$ are considered. First, subnetworks N_1, N_2, …, $N_{|N|-j}$ are merged into logical subnetwork $N_{1\sim|N|-j}$. The value of $Cut_{|N|-j}$ is then determined by procedure Test-and-Prune according to $p_i^{1\sim|N|-j+1}$ which has been determined in the $(j-1)$th iteration. Finally, the values of $p_i^{1\sim|N|-j}$ and $p_i^{|N|-j+1}$ are calculated. Internetwork data allocation phase of algorithm Layered-Cutting repeats the above steps until the value of Cut_1 has been determined. That is, after iterating $|N| - 1$ times, algorithm Layered-Cutting terminates inter-network data allocation phase and steps into intra-network data allocation phase. The algorithmic form of the procedure of inter-network data allocation phase in algorithm Layered-Cutting is as follows.

Procedure Inter-Network-Data-Allocation

1: Reorder all data items according to their $\sqrt{\frac{p_i}{l_i}}$ values in descending order
2: Let $p_i^{1\sim|N|}$ be p_i for each data item D_i
3: **for** (j = 1 to $|N| - 1$) **do**
4: Merge subnetworks N_1, N_2, …, $N_{|N|-j}$ into a logical subnetwork $N_{1\sim|N|-j}$
5: Employ procedure Test-and-Prune with *left* = 1 and *right* = $Cut_{|N|-j+1}$ to determine the value of $Cut_{|N|-j}$ according to $p_i^{1\sim|N|-j+1}$
6: Calculate $p_i^{1\sim|N|-j}$ for each data item D_i /* For next iteration */
7: Calculate $p_i^{|N|-j+1}$ for each data item D_i /* $p_i^{|N|-j+1}$ will be used in intra-network data allocation phase */
8: **end for**
9: **return** Cut_1, Cut_2, …, $Cut_{|N|-1}$

3.2.3 Determining values of cutting points

After describing the process of inter-network data allocation phase in algorithm Layered-Cutting, we now describe how to determine the values of cutting points in this subsection. Note that the determination method, called procedure Test-and-Prune, is used in algorithm Layered-Cutting described in section 3.2.2.

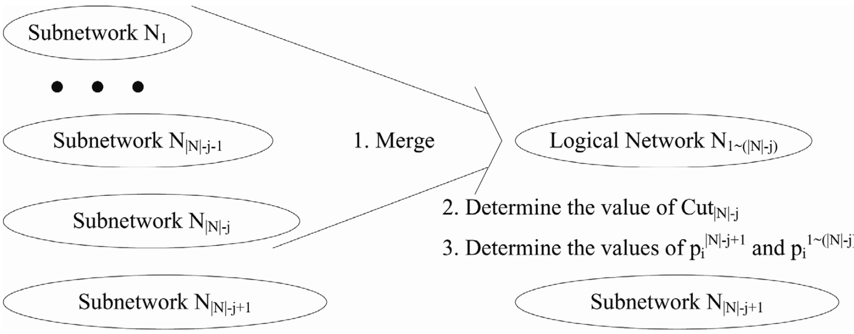


Fig. 3. Layered Cutting in the *j*th iteration.

We now consider the example of the *j*th iteration, and the example is shown in Fig. 3. In the *j*th iteration, we have to determine the value of $Cut_{|N|-j}$, where $1 \leq Cut_{|N|-j} \leq Cut_{|N|-j+1}$, so that the overall average access time of logical subnetwork $N_{1\sim|N|-j}$ and subnetwork $N_{|N|-j+1}$ is minimized.[3] To facilitate the following discussion, when $Cut_{|N|-j}$ is set to p, we denote the lower bound of the overall average access time of logical subnetwork $N_{1\sim(|N|-j)}$ obtained by Eq. (1) as $LB_{1\sim(|N|-j)}(p)$. Also let $LB_{|N|-j+1}(p)$ be the lower bound of subnetwork $N_{|N|-j+1}$ calculated by Eq. (1) or Eq. (2) as $LB_{|N|-j+1}(p)$.[4] As mentioned in section 3.2.1, the lower bound of overall average access time of the combination of subnetworks $N_1, N_2, \ldots, N_k$ can be approximated by the lower bound of the overall average access time of subnetwork $N_{1\sim k}$. Hence, the overall average access time of logical subnetwork $N_{1\sim|N|-j}$ and subnetwork $N_{1\sim|N|-j+1}$ when $Cut_{|N|-j} = p$ can be determined as $LB_{Cut|N|-j}(p) = w_{1\sim|N|-j}(p) \times LB_{1\sim(|N|-j)}(p) + w_{|N|-j+1}(p) \times LB_{|N|-j+1}(p)$, where $w_{1\sim|N|-j}(p)$ and $w_{|N|-j+1}(p)$ are the weights of logical subnetwork $N_{1\sim|N|-j}$ and subnetwork $N_{|N|-j+1}$, respectively, when $Cut_{|N|-j} = p$.

To obtain the optimal value of $Cut_{|N|-j}$, it is intuitive to scan all possible values of the cutting point and to select the best one as the value of $Cut_{|N|-j}$. However, this method is unscalable since the number of data items is usually large. In view of this, we design an efficient heuristic, named procedure Test-and-Prune, to determine a proper value of a cutting point in a "test-and-prune" manner. For better scalability, the objective of procedure Test-and-Prune is to find a local optimal value, instead of the optimal value, of $Cut_{|N|-j}$. Hence, we have the following definition.

Definition 5 A value of the cutting point, say p, is said *local optimal* if $LB_{Cut|N|-j}(p-1) > LB_{Cut|N|-j}(p)$, and $LB_{Cut|N|-j}(p+1) > LB_{Cut\{|N|-j}(p)$.

---

[3] The value of $Cut_{|N|}$ is defined as $|D|$.

[4] Eq. (1) is designed for a single-system network with *one* broadcast channel while Eq. (2) is for a single-system network with *multiple* broadcast channels.

Definition 6 A value of the cutting point, say q, is said to be *better* than another value of the cutting point, say p, if $LB_{Cut|N|-j}(q) < LB_{Cut|N|-j}(p)$.

The process of procedure Test-and-Prune is as follows. First, variables *left* and *right* are set, respectively, to the smallest and the largest possible values of $Cut_{|N|-j}$, and variable *middle* is set to $\left\lceil \frac{right + left}{2} \right\rceil$. Procedure Test-and-Prune checks whether setting $Cut_{|N|-j}$ to *middle* is local optimal. If so, procedure Test-and-Prune stops and suggests *middle* as the value of $Cut_{|N|-j}$. Otherwise, procedure Test-and-Prune checks the superiorities of setting $Cut_{|N|-j}$ to *middle* − 1, *middle* and *middle* + 1. If setting $Cut_{|N|-j}$ to *middle* − 1 is better than setting $Cut_{|N|-j}$ to *middle* and *middle* + 1, values from *middle* to $Cut_{|N|-j+1}$ are pruned and *right* is set to *middle* − 1. Otherwise, when setting $Cut_{|N|-j}$ to *middle* + 1 is better than setting $Cut_{|N|-j}$ to *middle* − 1 and *middle*, values from 1 to p are pruned and *left* is set to *middle* + 1. The above procedure repeats until a local optimal value of $Cut_{|N|-j}$ is found. The algorithmic form of procedure Test-and-Prune is as follows.

<u>**Procedure** Test-and-Prune(*min*, *max*)</u>
Parameters:
min: the smallest possible value of the cutting point
max: the largest possible value of the cutting point

left ← *min*
right ← *max*
middle ← $\left\lceil \frac{right + left}{2} \right\rceil$
while (*middle* is not better than *middle* − 1 and *middle* + 1) **do**
  if (*middle* − 1 is better than *middle* and *middle* + 1) **then**
    right ← *middle* − 1
  else /* *middle* + 1 is better than *middle* − 1 and *middle* */
    middle ← *middle* + 1
  end if
  middle ← $\left\lceil \frac{right + left}{2} \right\rceil$
end while
return *middle*

Due to the behavior of procedure Test-and-Prune, the worst case time complexity of procedure Test-and-Prune is $O(\log(right - left - 1)) = O(\log(right - left))$. Finally, we use the following example to illustrate the behavior of procedure Test-and-Prune.

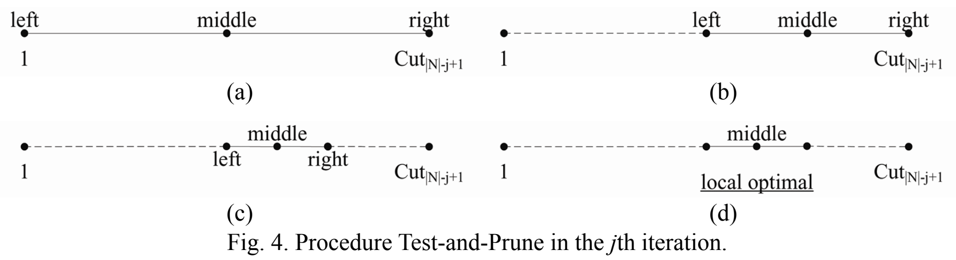


Fig. 4. Procedure Test-and-Prune in the jth iteration.

Example 1: Consider the example shown in Fig. 4 (a) that procedure Test-and-Prune is invoked to determine the value of $Cut_{|N|-j}$. Initially, *left* = 1, *right* = $Cut_{|N|-j+1}$, and *middle* is set to $\left\lceil \frac{right + left}{2} \right\rceil$. Procedure Test-and-Prune then tests whether setting $Cut_{|N|-j}$ to *middle* is better than setting $Cut_{|N|-j}$ to *middle* − 1 and *middle* + 1. Suppose that *middle* + 1 is better than *middle*. Then, as shown in Fig. 4 (b), values from 1 to *middle* are pruned. In addition, *left* is set to *middle* + 1 and *middle* is set to $\left\lceil \frac{right + left}{2} \right\rceil$.

Suppose that in the case shown in Fig. 4 (b), *middle* − 1 is better than *middle* and *middle* + 1. Values from *middle* to *right* are pruned, and *right* is set to *middle* − 1. Then *middle* is set to $\left\lceil \frac{right + left}{2} \right\rceil$, and the status is shown in Fig. 4 (c). In the case shown in Fig. 4 (c), suppose that *middle* is better than *middle* − 1 and *middle* + 1. Therefore, as shown in Fig. 4 (d), *middle* is local optimal and is reported as the suggested value of $Cut_{|N|-j}$. ❑

3.3 Phase Two: Intra-Network Data Allocation Phase

The objective of intra-network data allocation phase is to determine the broadcast programs of all subnetworks according to the resultant cutting configuration. It is intuitive that generating the broadcast program of subnetwork N_j is equivalent to executing algorithm BPG-Single on subnetwork N_j with data access probabilities $p_1^j, p_2^j, \ldots, p_{|D|}^j$. The algorithmic of intra-network data allocation phase in algorithm Layered-Cutting is as follows.

<u>**Procedure** Intra-Network-Data-Allocation</u>

1: **for** (j = 1 to $|N|$) **do**

2: Execute algorithm BPG-Single on subnetwork N_j according to p_i^j for each data item D_i to generate the broadcast program of subnetwork N_j /* The value of p_i^j for each data item D_i has already been calculated in procedure Inter-Network-Data-Allocation */

3: **end for**

4: **return** The broadcast programs of all subnetworks

Finally, the process of algorithm Layered-Cutting is to execute the procedures in inter-network data allocation and intra-network data allocation sequentially, and the algorithmic form of algorithm Layered-Cutting is as follows.

<u>**Algorithm** Layered-Cutting</u>

1: Execute procedure Inter-Network-Data Allocation

2: Execute procedure Intra-Network-Data Allocation

3: **return** The broadcast programs of all subnetworks returned by procedure Intra-Network-Data Allocation

3.4 Time Complexity Analysis

The worst case occurs when $|Cut_i| = |D|$ for i = 1, 2, …, $|N| - 1$. In the worst case, procedure Test-and-Prune will be executed $|N| - 1$ times with parameters *left* = 1 and *right* = $|D|$ in inter-network data allocation phase. Since the time complexity of procedure Test-and-Prune is $O(\log|D| - 1) = O(\log|D|)$, the time complexity of inter-network data allocation phase is $O((|N| - 1)\log|D|) = O(|N|\log|D|)$. In intra-network data allocation phase, algorithm BPG-Single is executed to generate the broadcast programs of all subnetworks. Hence, algorithm BPG-Single will be executed $|N|$ times, and the time complexity of intra-network data allocation phase is $O(|N|) \times O(BPG_{Single})$. Finally, the worst case time complexity of algorithm Layered-Cutting can be determined as the the summation of the time complexities of inter-network and intra-network data allocation phases, and is equal to $O(|N| \times (\log|D| + O(BPG_{Single})))$.

4. PERFORMANCE EVALUATION

According to the complexity analysis in section 3, algorithm Layered-Cutting is more efficient than algorithm Brute-Force at the cost of generating results worse than algorithm Brute-Force. Hence, we conduct several experiments to evaluate the performance of both algorithms and experimental results (including the execution time and the quality of results) are shown in the following subsections.

4.1 Simulation Model

To evaluation the performance of algorithm Layered-Cutting, we implement both algorithm Layered-Cutting and algorithm Brute-Force using C++. Similar to [13], we assume that the access probabilities of all data items follow a Zipf distribution with parameter θ. That is, the data access probability of data item D_i is equal to

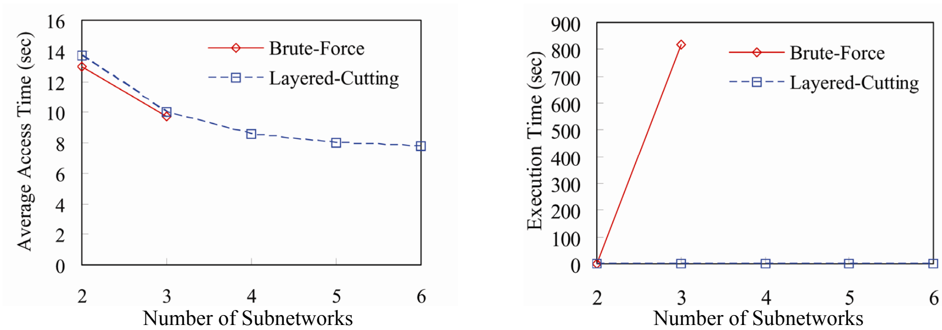
$$p_i = \frac{(\frac{1}{i})^\theta}{\sum_{j=1}^{|D|}(\frac{1}{j})^\theta}.$$

The default value of θ is set to be 0.8 with a reference to the analysis of real Web traces [23, 24]. Similar to [25], we assume that there are 1,000 data items and the data sizes are assumed to follow a normal distribution with mean 1 KBbytes.

In the model of subnetworks, we assume there are $|N|$ subnetworks in the multi-system heterogeneous overlayed network, and the service provider allocates three channels in each subnetwork for broadcasting data items. We also assume that subnetwork $|N|$ is able to cover the whole service area of the multi-system network and the rate between the sizes of the service areas of subnetwork i and subnetwork $i + 1$ is equal to 0.8. That is, $\frac{A_i}{A_{i+1}} = 0.8$ for all $1 \le i \le |N| - 1$. In addition, the ratio between the bandwidth of one broadcast channel in subnetwork i and that in subnetwork $i + 1$ is set to 2 (*i.e.*, $\frac{B_i}{B_{i+1}} = 2$), and $B_{|N|}$ is set to 10Kbytes [25]. We use the broadcast program generation algorithm proposed in [20] as algorithm BPG-Single. That is, the algorithm proposed in [20] is used to generate broadcast programs for all subnetworks in intra-network data allocation phase. For better readability, the default values of system parameters are listed in Table 2.

Table 2. System parameters.

Parameter	Default value
Number of subnetworks ($\|N\|$)	5
Number of broadcast channels in subnetwork $N_i(C_i)$	3
Number of data items($\|D\|$)	1000
Distribution of data item size (s_i)	normal with mean 1 KBytes
Distribution of data access probabilities (p_i)	Zipf with $\theta = 0.8$
Broadcast program generation for each subnetwork (algorithm BPG-Single)	The algorithm proposed in [20]



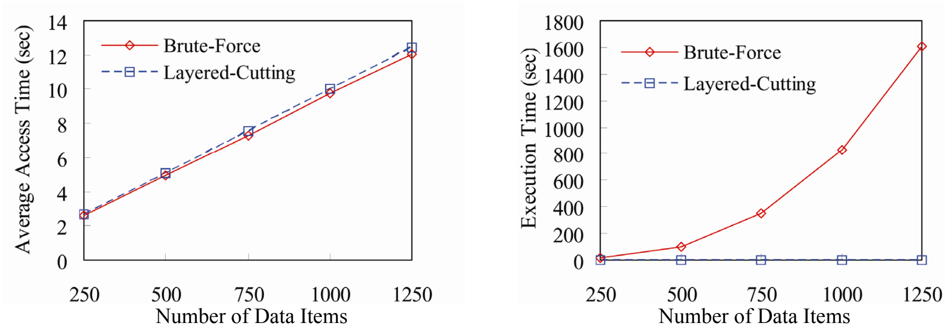
(a) Average access time. (b) Execution time.

Fig. 5. Impact of the number of subnetworks.

4.2 Impact of Number of Subnetworks

Fig. 5 shows the quality of solutions and execution time of algorithm Brute-Force and algorithm Layered-Cutting with the number of subnetworks varied. In this experiment, the number of subnetworks is setting from two to six. It is intuitive that increasing the number of subnetworks will decrease the average access time of the resultant broadcast programs since the total bandwidth of the multi-network system increases. Owing to the reason that networks with higher bandwidth usually cover smaller service area that those with lower bandwidth, the decrement of average access time decreases as the number of subnetworks increases. In this subsection, we only execute algorithm Brute-Force in the cases with two and three subnetworks because the execution time of algorithm Brute-Force in the case with four subnetworks is too longer (beyond one hour).

As observed in Fig. 5 (b), algorithm Brute-Force does not scale well. Such result conforms to the time complexity analysis in section 2.3. On the other hand, as shown in Figs. 5 (a) and (b), algorithm Layered-Cutting is able to obtain solutions close to optimal ones quickly. In this experiment, the degradation of solutions of algorithm Layered-Cutting over solutions of algorithm Brute-Force is smaller than 6%, and algorithm Layered-Cutting can terminate within one second. Such result also conforms to the time complexity of algorithm Layered-Cutting shown in section 3.4, and shows the advantage of algorithm Layered-Cutting.



(a) Average access time.

(b) Execution time.

Fig. 6. Impact of the number of data items.

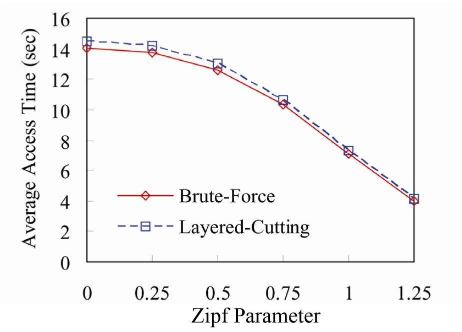
4.3 Impact of Number of Data Items

This experiment investigates the impact of the number of data items by setting the number of data items from 250 to 1250. The quality of resultant broadcast programs of algorithm Brute-Force and algorithm Layered-Cutting is shown in Fig. 6 (a). As observed, the solutions generated by algorithm Layered-Cutting is much close to those generated by algorithm Brute-Force (*i.e.*, optimal solutions) even the number of data items is set to 1250. In this experiment, the degradation of solutions of algorithm Layered-Cutting over solutions of algorithm Brute-Force is smaller than 4%.

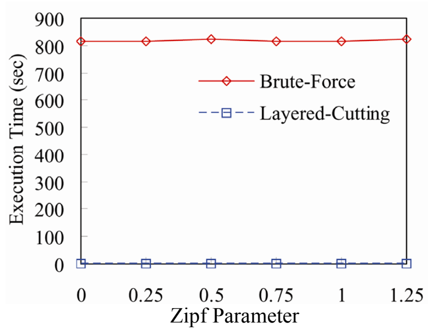
Fig. 6 (b) shows the execution time of both algorithms with the number of data items varied. It is intuitive that the execution time increases as the number of data items increases. According to the analysis in section 2.3, the time complexity of algorithm Brute-Force is much sensitive on the number of data items than algorithm Layered-Cutting. Such analytical result can be observed in Fig. 6 (b). The execution time of algorithm Brute-Force increases drastically as the number of data items increases. Under the same case, the execution time of algorithm Layered-Cutting only increases smoothly and can be terminated within one second. This result shows that algorithm Layered-Cutting is much scalable than algorithm Brute-Force.

4.4 Impact of Skewness of Data Access Probabilities

We now measure the impact of skewness of data access probabilities on the quality of resultant broadcast programs and execution time of both algorithms. The value of θ is set from zero to 1.25. Note that $\theta = 0$ indicates that the data access probability of each data item is equal. As shown in Fig. 7 (a), the average access time of the resultant broadcast programs of both algorithms greatly decreases as the value of θ increases. It is because when data access probabilities are highly skewed, several data items are requested frequently. Therefore, one transmission of one hot data item is of high probability to serve more data requests, thereby reducing overall average access time. In this experiment, the degradation of solutions of algorithm Layered-Cutting over solutions of algorithm Brute-Force is around 3%. As observed in Fig. 7 (b), the change of skewness of access probabilities only slightly affects the execution time of both algorithms.

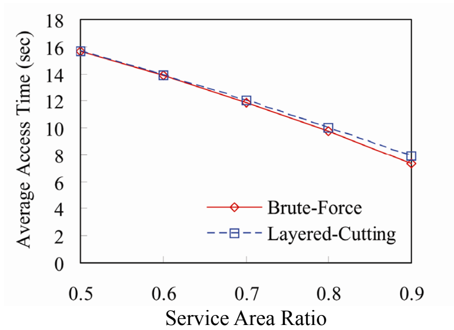


(a) Average access time.

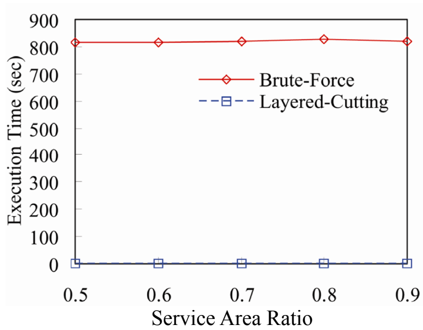


(b) Execution time.

Fig. 7. Impact of skewness of data access probabilities.



(a) Average access time.



(b) Execution time.

Fig. 8. Impact of service area ratio.

4.5 Impact of Service Area Ratio

Fig. 8 shows the quality of the resultant broadcast programs and execution time of both algorithms with service area ratio varied. As observed in Fig. 8 (b), the change of skewness of access probabilities only slightly affects the execution time of both algorithms. Considering Fig. 8 (a), it is obvious that increasing service area ratio will decrease overall average access time since subnetworks of higher bandwidth can cover larger service area and serve more data requests when service area ratio becomes large. We also observe from Fig. 8 (a) that the degradation of solutions of algorithm Layered-Cutting over solutions of algorithm Brute-Force increases from 0.2% to 8.3% as service area ratio increases from 0.5 to 0.9. It is due to the fact that algorithm Layered-Cutting is greedy-based and may miss some solutions with higher quality, and such effect becomes significant in the cases with large service area ratio. Fortunately, the quality of broadcast programs of algorithm Layered-Cutting is still close to those of algorithm Brute-Force even when service area ratio is set to 0.9. In addition, since being more faster than algorithm Brute-Force, algorithm Layered-Cutting is more suitable for practical use than algorithm Brute-Force.

5. CONCLUSION AND FUTURE WORK

A multi-system network consisting of multiple subnetworks is deemed to be able to provide better services by combining the relative merits of heterogeneous communication technologies. In addition, data broadcast is a promising technique to develop high scalability and energy-conserved mobile information systems. In this paper, we employed data broadcast in a multi-system heterogeneous overlayed wireless network and proposed a two-phase algorithm, named algorithm Layered-Cutting, to address the problem of broadcast program generation in a multi-system heterogeneous overlayed wireless networks. The experimental results showed that algorithm Layered-Cutting is able to efficiently generate broadcast programs of high quality for a multi-system heterogeneous overlayed wireless network.

Our future work is as follows. Since in practice, the service area of one subnetwork usually only partially overlap with that of another, we will consider in the future these cases and try to extend algorithm Layered-Cutting for such environments. One possible approach is to extend the formulae of p_i^j and $p_i^{1\sim j}$ to measure the load of each subnetwork where subnetworks may partially overlap one another. By applying the revised formulae, algorithm Layered-Cutting can directly solve the problem of broadcast program generation. Moreover, we will also try to develop a dedicated broadcast program generation algorithm for such environments. In addition, we will also take users' preferences on choosing subnetworks into consideration to make our algorithm more practicable.

APPENDIX

Determining the Values of p_i^j and $p_i^{1\sim j}$

Consider the case that the whole multi-system network receives f data requests and these data requests are assumed to be located in the service area of the whole multi-system network uniformly. We can expect there are $p_i \times f$ data requests for D_i. Let f^k_i and $f_i^{1\sim k}$ be the numbers of data requests for D_i served by subnetwork N_k and by logical subnetwork $N_{1\sim|k|}$, respectively. Initially, the values of all cutting points are undetermined. Since all requests of all data items must be served by the logical subnetwork $N_{1\sim|N|}$, we have $p_i^{1\sim|N|} = p_i$.

Consider the case that the value of $Cut_{|N|-1}$ has been determined. The whole system can be viewed as logical subnetwork $N_{1\sim|N|-1}$ and subnetwork $N_{|N|}$. For a data request for D_i, the data request will be served by the logic subnetwork $N_{1\sim|N|-1}$ only when when (1) $1 \le i \le Cut_{|N|-1}$ and (2) the user issuing this request is located in the service area of the logical subnetwork $N_{1\sim|N|-1}$. Therefore, all data requests for D_i, $i > Cut_{|N|-1}$, are served by subnetwork $N_{|N|}$. Suppose that all data requests are uniformly spread in the service area of the multi-system network (*i.e.*, the service area of subnetwork $N_{|N|}$). Then, $\frac{A_{|N|-1}}{A_{|N|}}$ data requests for D_i, $1 \le i \le Cut_{|N|-1}$ are served by logical subnetwork $N_{1\sim|N|-1}$ and the rest data requests are served by subnetwork $N_{|N|}$. As a consequence, we have

$$f_i^{|N|} = \begin{cases} \left(1 - \dfrac{A_{|N|-1}}{A_{|N|}}\right) \times p_i \times f, & \text{if } 1 \le i \le Cut_{|N|-1} \\ p_i \times f, & \text{otherwise} \end{cases}, \text{ and}$$

$$f_i^{1\sim|N|-1} = \begin{cases} \left(\dfrac{A_{|N|-1}}{A_{|N|}}\right) \times p_i \times f, & \text{if } 1 \le i \le Cut_{|N|-1} \\ 0, & \text{otherwise} \end{cases}.$$

By the definitions of p_i^j and $p_i^{1\sim j}$, we have

$$p_i^{|N|} = \frac{f_i^{|N|}}{\sum_{m=1}^{|D|} f_m^{|N|}}, \text{ and}$$

$$p_i^{1\sim|N|-1} = \frac{f_i^{1\sim|N|-1}}{\sum_{m=1}^{|D|} f_m^{1\sim|N|-1}}.$$

Therefore, after the value of $Cut_{|N|-1}$ has been determined, $f_i^{|N|}, f_i^{1\sim|N|-1}, p_i^{|N|}$ and $p_i^{1\sim|N|-1}$ can be obtained accordingly.

Now, consider the case of determining Cut_j. According to the process of algorithm Layered-Cutting, when determining Cut_j, the server have already determined the values of Cut_{j+1}, …, $Cut_{|N|-1}$ and the values of $f_i^{1\sim j+1}$, $1 \le i \le Cut_{j+1}$. Similarly, $\frac{A_j}{A_{j+1}}$ data requests from $f_i^{1\sim j+1}$, $1 \le i \le Cut_j$, are served by logical subnetwork $N_{1\sim j}$ and the rest data requests are served by subnetwork N_{j+1}. As a consequence, we have

$$f_i^{j+1} = \begin{cases} \left(1 - \dfrac{A_j}{A_{j+1}}\right) \times f_i^{1\sim j+1}, & \text{if } 1 \le i \le Cut_j \\ f_i^{1\sim j+1}, & \text{otherwise} \end{cases}, \text{ and}$$

$$f_i^{1\sim j} = \begin{cases} \left(\dfrac{A_j}{A_{j+1}}\right) \times f_i^{1\sim j+1}, & \text{if } 1 \le i \le Cut_j \\ 0, & \text{otherwise} \end{cases}.$$

By the definitions of p_i^j and $p_i^{1\sim j}$, we have

$$p_i^{j+1} = \frac{f_i^{j+1}}{\sum_{m=1}^{|D|} f_m^{j+1}}, \text{ and}$$

$$p_i^{1\sim j} = \frac{f_i^{1\sim j}}{\sum_{m=1}^{|D|} f_m^{1\sim j}}.$$

With the above results, the values of p_i^j and $p_i^{1\sim j}$ can be calculated recursively.

REFERENCES

1. R. Berezdivin, R. Breinig, and R. Topp, "Next-generation wireless communications concepts and technologies," *IEEE Communications Magazine*, Vol. 40, 2002, pp. 108-116.
2. G. Wu and M. Mizuno, "MIRAI architecture for heterogeneous network," *IEEE Communications Magazine*, Vol. 40, 2002, pp. 126-134.
3. S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks: data management for asymmetric communication environments," in *Proceedings of the ACM SIGMOD Conference*, 1995, pp. 198-210.
4. T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Data on air: organization and access," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, 1997, pp. 353-372.
5. W. C. Peng and M. S. Chen, "Efficient channel allocation tree generation for data broadcasting in a mobile computing environment," *ACM/Kluwer Wireless Networks*, Vol. 9, 2003, pp. 117-129.
6. S. Acharya and S. Muthukrishnan, "Scheduling on-demand broadcasts: new metrics and algorithms," in *Proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking*, 1998, pp. 43-94.
7. M. Agrawal, A. Manjhi, N. Bansal, and S. Seshan, "Improving web performance in broadcast-unicast networks," in *Proceedings of the IEEE INFOCOM Conference*, Vol. 1, 2003, pp. 229-239.
8. D. Aksoy and M. J. Franklin, "Scheduling for large-scale on-demand data broadcasting," in *Proceedings of IEEE INFOCOM Conference*, Vol. 2, 1998, pp. 651-659.
9. M. S. Chen, K. L. Wu, and P. S. Yu, "Optimizing index allocation for sequential data Broadcasting in wireless mobile computing," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, 2003, pp. 161-173.
10. S. C. Lo and A. L. P. Chen, "Optimal index and data allocation in multiple broadcast channels," in *Proceedings of the 16th International Conference on Data Engineering*, 2000, pp. 293-702.
11. J. L. Xu, B. Zheng, W. C. Lee, and D. K. Lee, "Energy efficient index for querying location-dependent data in mobile broadcast environments," in *Proceedings of the 19th International Conference on Data Engineering*, 2003, pp. 239-250.
12. B. Zheng, J. Xu, W. C. Lee, and D. L. Lee, "Energy-conserving air indexes for nearest neighbor search," in *Proceedings of the 9th International Conference on Extending Database Technology*, 2004, pp. 48-66.
13. J. L. Huang, W. C. Peng, and M. S. Chen, "Binary interpolation search for solution mapping on broadcast and on-demand channels in a mobile computing environment," in *Proceedings of the 10th ACM International Conference on Information and Knowledge Management*, 2001, pp. 434-441.

14. W. C. Lee, Q. L. Hu, and D. L. Lee, "A study on channel allocation for data dissemination in mobile computing environments," *ACM/Kluwer Mobile Networks and Applications*, Vol. 4, 1999, pp. 117-129.
15. J. L. Huang and M. S. Chen, "Dependent data broadcasting for unordered queries in a multiple channel mobile environment," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, 2004, pp. 1143-1156.
16. J. L. Huang, M. S. Chen, and W. C. Peng, "Broadcast dependent data for ordered queries without replication in a multi-channel mobile environment," in *Proceedings of the 19th IEEE International Conference on Data Engineering*, 2003, pp. 692-694.
17. G. Lee and S. C. Lo, "Broadcast data allocation for efficient access of multiple data items in mobile environments," *ACM/Kluwer Mobile Networks and Applications*, Vol. 8, 2003, pp. 365-375.
18. G. Lee, S. C. Lo, and A. L. P. Chen, "Data allocation on the wireless broadcast channel for efficient query processing," *IEEE Transactions on Computers*, Vol. 51, 2002, pp. 1237-1252.
19. M. Stemm and R. H. Katz, "Vertical handoffs in wireless overlay networks," *ACM/Kluwer Mobile Networks and Applications*, Vol. 3, 1998, pp. 319-334.
20. N. H. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," *ACM/Kluwer Wireless Networks*, Vol. 5, 1999, pp. 171-182.
21. K. Prabhakara, K. A. Hua, and J. H. Oh, "Multi-level multi-channel air cache designs for broadcasting in a mobile environment," in *Proceedings of the 16th International Conference on Data Engineering*, 2000, pp. 167-186.
22. W. G. Yee, S. B. Navathe, E. Omiecinski, and C. Jermaine, "Efficient data allocation over multiple channels at broadcast servers," *IEEE Transactions on Computers*, Vol. 51, 2002, pp. 1231-1236.
23. L. Breslau, P. Cao, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proceedings of IEEE INFOCOM Conference*, Vol. 1, 1999, pp. 126-134.
24. V. Padmanabhan and L. Qiu, "The content and access dynamics of a busy web site: Findings and implications," in *Proceedings of IEEE SIGCOMM Conference*, Vol. 30, 2000, pp. 111-123.
25. C. W. Lin, H. Hu, and D. L. Lee, "Adaptive realtime bandwidth allocation for wireless data delivery," *ACM/Kluwer Wireless Networks*, Vol. 10, 2004, pp. 103-120.

Jiun-Long Huang (黃俊龍) received his B.S. and M.S. degrees in Computer Science and Information Engineering Department in National Chiao Tung University in 1997 and 1999, respectively, and his Ph.D. degree in Electrical Engineering Department in National Taiwan University in 2003. Currently, he is an assistant professor in Computer Science Department in National Chiao Tung University. His research interests include mobile computing, wireless networks, and data mining.

Jui-Nan Lin (林瑞男) received his B.S. degree in Computer Science and Information Engineering Department in National Chiao Tung University in 2005. Currently, he is a graduate student in Institute of Network Engineering in National Chiao Tung University. His research interests include wireless sensor networks, and pervasive computing.