JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 17, 647-665 (2001)

Heuristic Algorithm for Optimal Design of the Two-Level Wireless ATM Network

DER-RONG DIN AND S. S. TSENG
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.

In this paper, we investigate the optimal assignment problem of cells in *PCS* (*Personal Communication Service*) to switches in a wireless ATM network. Given cells and switches on an ATM network (whose locations are fixed and known), the problem is grouping cells into clusters and assigning these clusters to the switches in an optimum manner. This problem is modeled as a complex integer programming problem and finding an optimal solution of this problem is *NP-complete*. A three-phase heuristic algorithm *MCMLCF* (Maximum cell and minimum local communication first) consisting of *Cell Pre-Partitioning Phase*, *Cell Exchanging Phase*, and *Cell Migrating Phase*, is proposed. First, in the Cell Pre-Partitioning Phase, a three-step procedure (*Clustering Step, Packing Step, and Assigning Step*) is proposed to group cells into clusters. Second, Cell Exchanging Phase is proposed to greatly improve the result by repeatedly exchanging two cells in different switches. Finally, Cell Migrating Phase is proposed to reduce cost by repeatedly migrating all cells in a used switch to an empty switch. Experimental results indicate that the proposed algorithm runs efficiently. Comparing the results of the algorithm to a naive heuristic called *NSF*, we have shown that the computation time is reduced by 30.1%. Experimental results show that Cell Exchanging and Cell Migrating phases can reduce the total cost by 34.1% on average. By comparing the results of the proposed algorithm to the genetic algorithm, the heuristic method came close to optimum - on average within 5%.

Keywords: wireless ATM, design of algorithms, assignment problem, clustering problem, graph partitioning problem, personal communication services

1. INTRODUCTION

Recently there has been much interest in extending ATM technology into the wireless environment [1-11]. The motivation behind this (termed *wireless ATM*) includes the desire for seamless interconnection of wireless and ATM networks, and the need to support emerging mobile multimedia services. However, due to inherent differences in these two types of networks, the introduction of ATM into the wireless environment presents many interesting challenges [3, 10]. These include supporting an end-to-end ATM connection with user mobility, handling the high error rate performance of wireless links and grouping cells into LAs (location areas or clusters).

In this paper, we investigate the optimal assignment of cells in PCSs (Personal Communication Services) to switches in an ATM network. The system area of PCSs is

---

Received December 1, 1999; revised June 1 & July 10, 2000; accepted August 9, 2000.
Communicated by Norio Shiratori.

divided into several *LAs* (*location areas*). In general, an LA consists of an aggregation of cells forming a contiguous geographical region. When a subscriber enters a cell that belongs to a different LA, a *location update* (*LU*) procedure that informs the network about the subscriber's new location is performed. This will generate network traffic overhead in PCS networks and consume scarce radio resources. Moreover, LU also increases the load on distributed location databases and, thus, increases the complexity of implementing the databases [7, 9].

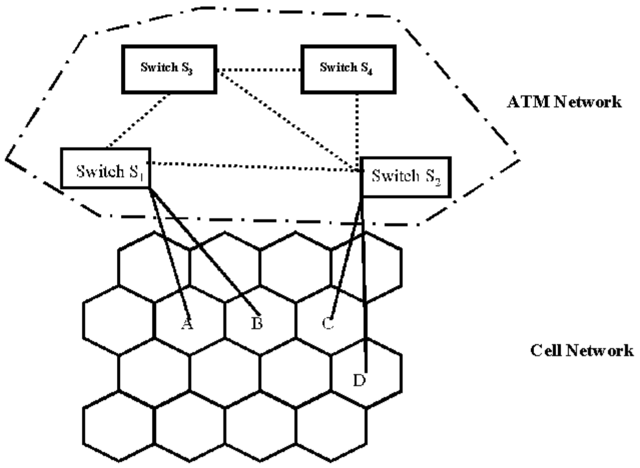

Fig. 1. Two-level hierarchical network. The handoff from *B* to *C* is more expensive than from *B* to *A*.

Consider the example shown in Fig. 1, where cells *A* and *B* are connected to switch s_1, and cells *C* and *D* are connected to switch s_2. If the subscriber moves from cell *B* to cell *A*, switch s_1 will perform a handoff for this call. This handoff is relatively simple and does not involve any location update in the databases that record the position of the subscriber. The handoff also does not involve any network entity other than switch s_1. Now, let us suppose that the subscriber moves from cell *B* to cell *C*. Then the handoff involves the execution of a fairly complicated protocol between switches s_1 and s_2. In addition, the location of the subscriber in the databases must be updated. There is actually one more fact that makes this type of handoff difficult. If switch s_1 is responsible for keeping the billing information about the call then, switch s_1 cannot simply remove itself from the connection as a result of the handoff. In fact, the call continues to be routed through switch s_1 (for billing purposes). In this case, the connection is from cell *C* to switch s_2, then to switch s_1, and finally to the telephone network [10].

Merchant and Sengupta [10] considered the problem of assigning cells to switches in PCS network. They formulated the problem and proposed a heuristic algorithm to solve it so that the total cost can be minimized. The total cost consists of cabling and location update. The location update cost considered in [10], which depends only on the frequency of handoff between two switches, is not practical. Since the switch of the ATM backbone is wide spread, the communication cost between the two switches should be considered in calculating the location update cost.

In this paper, we are given a group of cells and a group of switches in an ATM network (whose locations are fixed and known). The problem is to group cells into LAs and assign LAs to switches in the ATM network in an optimum manner. We consider the topological design of a two-level hierarchical network. The upper-level of the network is a connected ATM network, while the lower-level is a cell network which is configured as an H-mesh (see Fig. 1). The objective cost has two components. One is the location update cost, which involves two different switches; the other is the cost of cabling which connected cells to switches of the ATM network. We try to assign cells to switches so that the total cost is minimized under some assumptions described later. The organization of this paper is shown as follows. In Section 2, we formally define the problem. In Section 3, we outline the proposed algorithms. In Sections 4, 5 and 6, we describe details of the algorithms. The experimental results are presented in Section 7. Finally, a conclusion is given in Section 8.

2. PROBLEM FORMULATION

This section first provides an overview of various terms and notations used to explain the concepts outlined in the subsequent sections. Then, we introduce the definitions of various parameters used in this paper, and list assumptions made in representing the network topology.

Notation:

n = total number of cells in the PCS network

m = total number of switches in the ATM network

$G(S, E)$ = ATM network, where S is the set of switches and $E \subseteq S \times S$

$CG(C, L)$ = cell network, where C is the set of cells and $L \subseteq C \times C$

(s_k, s_l) = edge between switches s_k and s_l in S

(c_i, c_j) = edge between cell c_i and c_j in C

$(X_{s_k}, Y_{s_k},)$ = coordinate of switch $s_k \in G$, $k = 1, 2, \ldots, m$

(X_{c_i}, Y_{c_i}) = coordinate of cell $c_i \in C$, $i = 1, 2, \ldots, n$

d_{kl} = minimum communication cost between switches s_k and s_l in G

f_{ij} = frequency per unit time of the handoffs that occur between cells c_i and c_j in CG, $i, j = 1, \ldots, n$

l_{ik} = cost of cabling per unit time between cell $c_i \in CG$ and switch $s_k \in G$, $i = 1, \ldots, n$; $k = 1, \ldots, m$; assume

$$l_{ik} = \sqrt{(X_{c_i} - X_{s_k})^2 + (Y_{c_i} - Y_{s_k})^2}$$

w_{ij} = weight of edge $(c_i, c_j) \in CG$ where $w_{ij} = f_{ij} + f_{ji}$, $w_{ij} = w_{ji}$, and $w_{ii} = 0$; $i, j = 1, \ldots, n$

Cap = cell handling capacity of the switch

$m' = \left\lceil \dfrac{n}{Cap} \right\rceil$ = number of switches that need to be assigned

α = ratio of cabling cost to handoff cost

Assumptions:

(1) The structures and positions of the ATM network and PCS networks are known.

(2) We assume that the cost of handoffs involving only one switch is negligible.

(3) Each cell in the cell network will be directly assigned and connected to only one switch in the ATM network.

(4) The capacity of a switch, defined as the number of cells that it can handle, is limited to a maximum *Cap*.

(5) The cost has two components, the cost of handoffs that involve two switches, and the cost of *cabling* (or *trucking*).

(6) *minimal switches assumption*: It is worth noting that if the number of assigned switches is minimized, then the location update (handoff) costs will be reduced.

(7) *load balance assumption*: The load on assigned switches is assumed to be balanced. If the load balance assumption is satisfied, $m' = \left\lceil \frac{n}{Cap} \right\rceil$ switches need to be assigned, and the number of cells assigned to switches is $\left\lfloor \frac{n}{m'} \right\rfloor$ or $\left\lceil \frac{n}{m'} \right\rceil$.

Decision variables:

$$x_{ik} = \begin{cases} 1 & \text{if cell } c_i \text{ is assigned to switch } s_k \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ijk} = x_{ik} x_{jk}, \text{for } i, j, = 1, \ldots, n \text{ and } k = 1, \ldots, m$$

$$i.e., z_{ijk} = \begin{cases} 1 & \text{if both cells } c_i \text{ are connected to a common switch } s_k \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} = \sum_{k=1}^{m} z_{ijk}, i, j = 1, \ldots, n$$

$$i.e., y_{ij} = \begin{cases} 1 & \text{if both cells } c_i \text{ and } c_j \text{ are connected to a common switch} \\ 0 & \text{otherwise} \end{cases}$$

Problem:

Find variables x_{ij} which minimize

$$\sum_{i=1}^{n} \sum_{k=1}^{m} l_{ik} x_{ik} + \alpha \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{m} w_{ij} (1 - y_{ij}) x_{ik} x_{jl} d_{lk} \tag{1}$$

subject to

$$\sum_{i=1}^{n} x_{ik} \le Cap, k = 1, \ldots, m; \tag{2}$$

$$\sum_{k=1}^{m} x_{ik} = 1, \text{ for } i = 1, \ldots, n; \tag{3}$$

$$\left\lfloor \frac{n}{m'} \right\rfloor \le \sum_{i=1}^{n} x_{ik} \le \left\lceil \frac{n}{m'} \right\rceil, k = 1, \ldots, m; \tag{4}$$

$$x_{ik} \in \{0,1\} \text{for } i = 1, \ldots, n; \text{for } k = 1, \ldots, m. \tag{5}$$

If cells c_i and c_j are assigned to different switches, then a cost is incurred. Every time a handoff occurs; if f_{ij} is the frequency per unit time of a handoff that occurs between cells c_i and c_j, ($i, j = 1,..., n$), which we assume, is fixed and known. Our objective is to assign each cell to a switch so as to minimize the sum of the cabling and handoff costs per unit time, i.e., the total cost.

Thus, our goal is to minimize objective function (1). The first part of (1) is the total cabling cost between cells and switches, the second part is the cost of handoffs per unit time, and α is the ratio cabling cost to handoff cost. Constraint (2) ensures that the number of calls is limited to the capacity *Cap*. Constraint (3) ensures that each cell is assigned to exactly one switch. Constraint (4) ensures that the minimal used switches assumption and load balance assignment assumption are satisfied. Constraint (5) ensures that x_{ik} is a binary value.

3. OUTLINE OF HEURISTIC ALGORITHM

It is well known that finding an optimal solution to this problem is *NP-hard*. In this paper, a three-phase heuristic algorithm is proposed to solve this problem, which can be described as follows:

(1) *Cell Pre-Partitioning Phase*: Construct a pre-partition of *CG* by considering some basic constraints.
(2) *Cell Exchange Phase*: Take the ATM network environment into account, and improve of pre-partition by exchanging cells in different switches.
(3) *Cell Migration Phase*: Migrate cells in one switch to an empty switch such that the total cost can be reduced.

The algorithm's input are *CG*, *G* and parameters (α, *CAP*,...). The goal of Cell Pre-Partitioning Phase is to group cells in *CG* into clusters and to assign clusters to switches of ATM network *G*. It is worth noting that if two cells are connected to the same switch, the handoff cost between two cells can be ignored. Thus, cells are grouped into clusters in Clustering Step, based on some threshold values defined later. Since threshold constraints exist, after Clustering Step completing, the number of clusters may be greater than m', which is further merged and adjusted in Packing Step. In Assignment Step, there are m' clusters of *CG* that will be assigned to m switches of the ATM switch network *N*. This subproblem can be formulated as an assignment problem [13, 14].

After executing Cell Pre-Partitioning Phase, it can be easily seen that each cell is connected to one switch and the minimal switches and load balance assumptions are satisfied. Obviously, during Clustering Step, the transmission cost of the ATM network is ignored. The Assigning Step assigned clusters to switches and did not consider the cabling cost between switches and individual cells. As a result, Cell Exchanging Phase and Cell Migrating Phase were developed to improve the result of Cell Pre-Partitioning Phase. The main objective of Cell Exchanging Phase is to reduce the total cost by considering the 2-layer cell-switch network. Two cells in different switches which provide the greatest improvement are selected, and the assigned switches are exchanged. This process continuously runs until no more improvement can be made. In Cell Migrating Phase, a used switch and an empty switch which provide the greatest improvement are selected, and all cells the switch being used are migrated to the empty switch. This process also runs continuously until no more improvement can be made.

4. CELL PRE-PARTITIONING PHASE

In this section, the three steps that make up the Cell Pre-Partitioning Phase are described. They are:

(1) *Clustering Step*: Group cells into clusters so as to reduce the handoff cost between cells.
(2) *Packing Step*: Reduce the number of clusters to m'.
(3) *Assignment Step*: Assign clusters to switches so that the cabling cost is minimized.

Example 1. Consider the two graphs shown in Fig. 2. There are ten cells in *CG* which should be assigned to four switches in *S*. The edge weight between two cells is the frequency per unit time of the handoffs that occur between them. Four switches are positioned at the center of cells: c_1, c_5, c_7, and c_9. Assume the matrix *CS* of the distance between cells and switches is

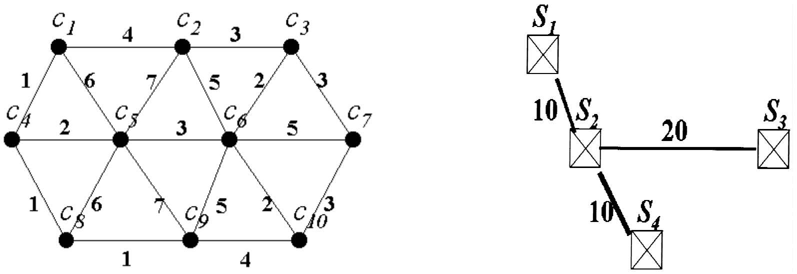

Fig. 2. Example of cell and ATM networks.

$$CS = \{l_{ik}\} = \begin{bmatrix} 0 & 1 & \sqrt{7} & 2 \\ 1 & 1 & \sqrt{3} & \sqrt{3} \\ 2 & \sqrt{3} & 1 & 2 \\ 1 & 1 & 3 & \sqrt{3} \\ 1 & 0 & 2 & 1 \\ \sqrt{3} & 1 & 1 & 1 \\ \sqrt{7} & 2 & 0 & \sqrt{3} \\ \sqrt{3} & 1 & \sqrt{7} & 1 \\ 2 & 1 & \sqrt{3} & 0 \\ \sqrt{7} & \sqrt{3} & 1 & 1 \end{bmatrix}$$

4.1 Clustering Step

In Clustering Step, three thresholds are introduced to constrain the clustering process.

(1) *Average load* ($avgLoad = \left\lceil \frac{n}{m'} \right\rceil$): This is used to avoid generating extremely large and load inbalanced clusters.
(2) *Average edge cost* ($avgEdgeCost = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} /(n^2 - n)$): This is used to avoid merging very low-cost edges.
(3) *Average cell to cell distance* ($avgCellDist = \sum_{i=1}^{n} \sum_{j=1}^{n} \sqrt{(X_{c_i} - X_{c_j})^2 + (Y_{c_i} - Y_{c_j})^2} /(n^2 - n)$) : This is used to avoid merging distant cells.

Let $W(c_i)$ be the load on c_i with initial value 1, and let $dist(i, j)$ be the Euclidean distance of two cells c_i and c_j. To decide whether or not two cells can be merged, three merging constraints are defined. If cells c_i and c_j are merged, the following constraints must be satisfied:

(1) $W(c_i) + W(c_j) \leq avgLoad$.
(2) $w_{ij} > avgEdgeCost$.
(3) $dist(i, j) \leq avgCellDist$.

Initially, set A is formed as the set of n cells. The clustering procedure is to iteratively merge a pair of cells which satisfy the merging criterion. A new cell is formed with its location at the *center-of-mass* (*COM*) of the two cells being merged, where the weights used in the COM calculation are simply the weights assigned to each cell. The group of real cells, which represent merging the two old cells, forms the new cell. The old cells are then removed from set A. The new cell will be added to set A later. The weight of the new cell is simply the sum of the weights of the old cells. Note that with the initial weights all set equal to one, the weight is simply the number of real cells represented by the new cell. The location update cost between the new cell and the old cells is the sum of the location update costs between the two old cells and the other cells.

Algorithm: Clustering

Step 1.1 Repeat Steps 1.2, 1.3 and 1.4 until no more cell can be merged.
Step 1.2 Find the cell *activecell* with maximal edge sum (edge radiated from it) in the cell graph.
Step 1.3 Find the cell *condcell* and the *edge* connected to *activecell* with minimum ‘radiating sum’ which satisfies the three merge constraints.
Step 1.4 If no suitable *condcell* is found, repeat Step 1.5 until no more cells can be merged. Otherwise, merge *condcell* to *activecell*, if (*condcell* < *activecell*); or vice versa.
Step 1.5 Find the next cell *activecell* with maximal edge sum (edge radiated from it) in the cell graph. Find the cell *condcell* and the edge connected to *activecell* with minimal ‘radiating sum’ which satisfies three merge constraints. If no suitable *condcell* found then algorithm terminate; otherwise merge *condcell* to *activecell*, if (*condcell* < *activecell*); or vice versa.

An example of applying Algorithm Clustering to Example 1 is shown in Fig. 3. First, select c_5 as *activecell* and c_8 as *condcell*. In Fig. 3(b), the two cells are merged into a new cell c_5^* in Step 1.4, and cell c_5^* is then selected as the new *activecell*. Repeating this process, the cell network can finally be clustered into the two groups shown in Fig. 3(i).

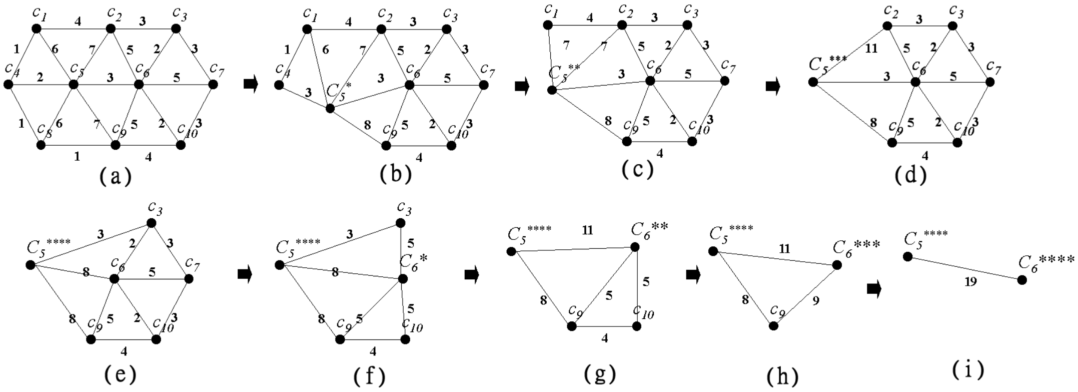

Fig. 3. Result of applying clustering algorithm to Example 1.

4.2 Packing Step

To satisfy the load balance assignment and minimal switches assumption, this step merges some small cells (clusters) left into the cell graph to larger cells in order to reduce the number of clusters to m'.

Algorithm: Packing

Step 2.1 Repeat Steps 2.2, 2.3 and 2.4 until the number of clusters remaining in *CG* is m'.
Step 2.2 Sort the remaining clusters in nondecreasing order according to the weight of each cluster (or cell).
Step 2.3 Select the minimal weight cluster c_i and find the cluster c_j with the greatest weight such that $W(c_i) + W(c_j) < avgLoad$. If found, merge the two clusters which decreases the number of clusters by 1. Otherwise, go to Step 2.4.
Step 2.4 For each cell in the minimal weight cluster c_i, find the heaviest weight edge connected to it and the corresponding cluster; merge the cell into this cluster. This process is repeated until all cells in this cluster are adjusted. Then, decreases the number of clusters by 1.

4.3 Assignment Step

After obtaining m' clusters, the goal of this step is to assign clusters to switches so as to minimize cabling costs between clusters and switches. Let $P = \{P_l\}$, $l = 1, \ldots, m'$, ($m' < m$) be an m'-way partition obtained from previous steps of *CG*. Define $F(k, l)$ as the sum of cabling costs of cells which are clustered into P_k and assigned to switch l, $1 \le k \le m'$, $1 \le l \le m$; i.e.,

$$F(k,l) = \sum_{c_i \in P_k} l_{il}, 1 \le k \le m', 1 \le l \le m.$$

Obviously, the subproblem can be formulated as a minimal weighted matching problem on a bipartite graph which is known as the *assignment problem*. Therefore, the Hitchcoch Algorithm [14] can be applied to find the optimal solution of the assignment problem in $O(m^3)$ time. The result after running Assigning Step is shown in Fig. 4. Cells c_1, c_2, c_4, c_5 and c_8 are connected to switch s_2 with cabling cost 4, while the others are connected to switch s_4 with cabling cost 5.732.

5. CELL EXCHANGING PHASE

The goal of Cell Exchanging Phase is to select two cells in different switches and exchange them in order to reduce the total cost. The basic idea of Cell Exchanging Phase is borrowed from Kernighan-Lin [15] algorithm which only works on traditional 2-way graph partition problems. In our 2-layer cell-switch network environment, location update costs and cabling costs must be considered simultaneously. Hence, we modified Kernighan-Lin's procedure to exchange cells in different clusters by selecting the "most preferable" cells to exchange instead of exchanging two arbitrary cells. We first introduce some notations required in the following. Given m' nonempty sets of cells $P = \{P_l\}$, $l = 1, 2, ..., m'$, P is called a m'-way partition of CG, if $P_1 \cup P_2 \cup ... \cup P_{m'} = C$ and $P_i \cap P_j = \phi$, where $i \neq j$. Without loss of generality, we assume the cells in set P_j are assigned to switch s_j, $j = 1, .., m'$. Let $sid(c_i) = l$ if c_i is in P_l, l is called the sid of cell c_i. Let $LUCS(i, l) = \sum_{j \in P_l} w_{ij}$ and $D(i, l) = LUCS(i, l) - LUCS(i, sid(c_i))$. Then for a given partition P, the location update cost of the partition is:

$$\alpha \sum_{c_i \in C} \sum_{s_l \in S} (LUCS(i,l) \cdot d_{sid(c_i)l})$$

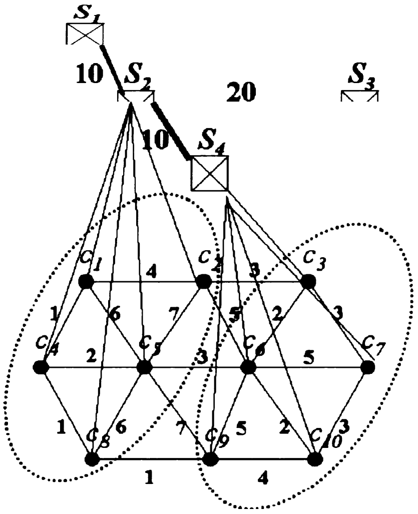

Fig. 4. Result of applying Assignment Step to Example 1.

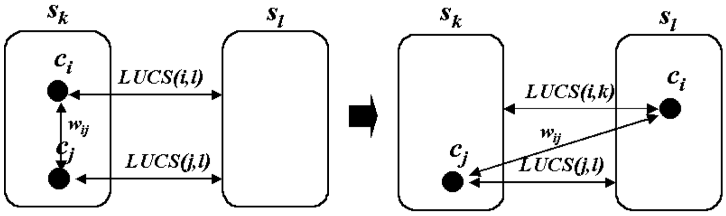

Fig. 5. Cell c_i is moved from switch s_k to s_l.

If cell c_i which is currently assigned to switch s_k, is reassigned to switch s_l, then the gain of reduced objective saving cost can be computed from the following lemma:

Lemma 1. Ignoring the size restriction, if cell c_i which is currently assigned to s_k, is moved to switch s_l, then

(1) the cabling cost is reduced by $d_{ik} - d_{il}$ and
(2) the location update cost is reduced by $\alpha(LUCS(i, l) - LUCS(i, k)) \cdot d_{kl}$

Proof: The proof of the cabling cost can be easily obtained. Let z be the sum of weights of all edges except those between cell c_i and all cells of P_l. Then from Fig. 5, we can see

old cost = $(z + LUCS(i, l)) \times d_{kl} \times \alpha$
new cost = $(z + LUCS(i, k)) \times d_{kl} \times \alpha$

The reduced location update cost (handoff cost)

$= (LUCS(i, l) - LUCS(i, k)) \times d_{kl} \times \alpha$
$= D(i, l) \times d_{kl} \times \alpha.$ ❑

Now, if cell c_i and c_j are exchanged, the gain can be computed from the following lemma:

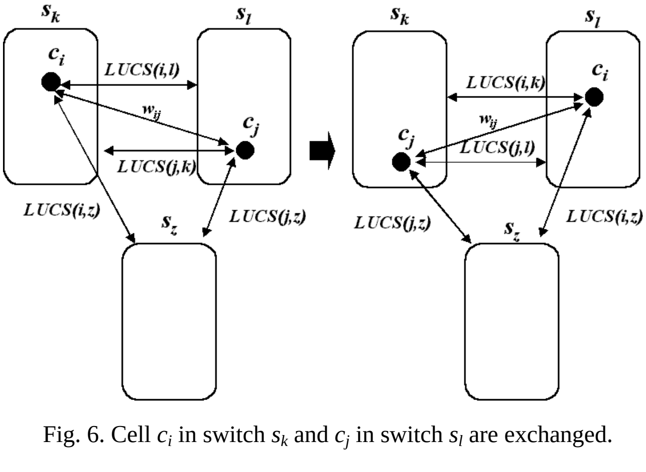

Fig. 6. Cell c_i in switch s_k and c_j in switch s_l are exchanged.

Lemma 2. If cell c_i in switch s_k and c_j in switch s_l are exchanged, then

(1) The reduced cabling cost is: $RC(i, j) = d_{ik} - d_{il} + d_{jl} - d_{jk}$
(2) The reduced location update cost is:

$$RLU(i, j) = (D(i, l) - D(j, k) - 2w_{ij}) \times d_{kl} \times \alpha + (d_{kz} - d_{lz}) \times \alpha \times (LUCS(i, s) - LUCS(j, z))$$

(3) The reduced total cost is: $exchange(i, j) = RC(i, j) + RLU(i, j)$.

Proof: The proofs of (1) and (3) are straightforward. From Lemma 1 and Fig. 6, the reduced cost of cell c_i is:

$$(LUCS(i, l) - LUCS(i, k) - w_{ij}) \times d_{kl} \times \alpha;$$

the reduced cell c_j cost is:

$$(LUCS(j, k) - LUCS(j, l) - w_{ij}) \times d_{kl} \times \alpha;$$

the reduced cost of switch s_z to cell c_i is ($z \neq k$ and $z \neq l$):

$$(d_{kz} - d_{lz}) \times \alpha \times LUCS(i, z).$$

the reduced cost of switch s_z to cell c_j is ($z \neq k$ and $z \neq l$):

$$(d_{lz} - d_{kz}) \times \alpha \times (LUCS(i, z) - LUCS(i, z)).$$

The reduced location update cost is:

$$(D(i, l) - D(j, k) - 2w_{ij}) \times d_{kl} \times \alpha + (d_{kz} - d_{lz}) \times \alpha \times (LUCS(i, z) - LUCS(j, z)). \quad \square$$

After cells c_i and c_j have been selected and exchanged, the matrix LUCS should be updated so that the algorithm can run effectively and avoid sequential searching.

Lemma 3. After cells c_i in switch s_k and c_j in switch s_l are exchanged, the *LUCS* values can be updated as follows:

(1) $LUCS^{new}(i, k) = LUCS(i, k) + w_{ij}$
(2) $LUCS^{new}(i, l) = LUCS(i, l) - w_{ij}$
(3) $LUCS^{new}(j, k) = LUCS(j, k) - w_{ij}$
(4) $LUCS^{new}(j, l) = LUCS(j, l) + w_{ij}$

For any cell a in switch s_z
(5) $LUCS^{new}(a, k) = LUCS(a, k) + w_{aj} - w_{ai}$, if ($s_z = k$) and ($a \neq i$)
(6) $LUCS^{new}(a, l) = LUCS(a, l) - w_{aj} + w_{ai}$, if ($s_z = k$) and ($a \neq i$)
(7) $LUCS^{new}(a, k) = LUCS(a, k) + w_{aj} - w_{ai}$, if ($s_z = l$) and ($a \neq j$)
(8) $LUCS^{new}(a, l) = LUCS(a, l) - w_{aj} + w_{ai}$, if ($s_z = l$) and ($a \neq j$)

(9) $LUCS^{new}(a, k) = LUCS(a, k) + w_{aj} - w_{ai}$, if ($s_z \neq k$ and $s_z \neq l$) and ($a \neq i$ and $a \neq j$)
(10) $LUCS^{new}(a, l) = LUCS(a, l) - w_{aj} + w_{ai}$, if ($s_z \neq k$ and $s_z \neq l$) and ($a \neq i$ and $a \neq j$)

Proof: The proofs of (1)-(4) are trivial. For a cell c_a, which is neither with sid c_i nor with sid c_j (see Fig. 7), $LUCS(a, k)$ is increased by $-w_{ai} + w_{aj}$(9), and $LUCS(a, l)$ value is increased by $-w_{aj} + w_{ai}$(10).

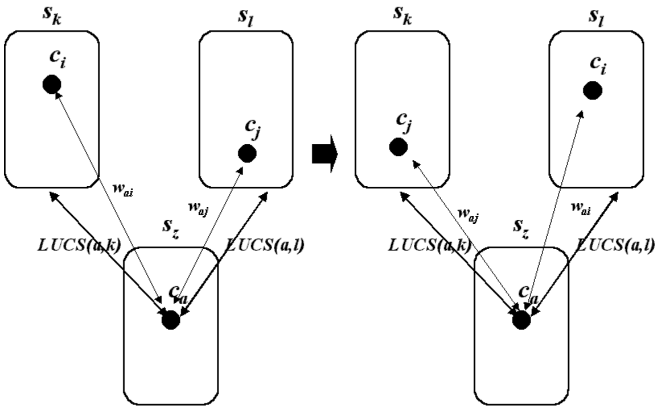

Fig. 7. Cell c_i in switch s_k and c_j in switch s_l are exchanged.

If $sid(c_a) = sid(c_i) = k$(see Fig. 8) then $LUCS(i, k)$ and $LUCS(i, l)$ are increased by $w_{aj} - w_{ai}$ and $w_{ai} - w_{aj}$, respectively. Thus, (5) and (6) hold. For cell c_a with sid l, a similar argument shows that (7) and (8) hold. ❑

Given an initial assignment, Cell Exchanging Phase consecutively selects two cells in different switches to exchange. At each iteration, two cells c_a and c_b are selected which maximize the reduced exchanging cost $exchange(a, b)$ where

$$exchange(a,b) = \max_{(c_i,c_j)\in C\times C} exchange(i, j).$$

The iteration continues if $exchange(a, b) \geq 0$.

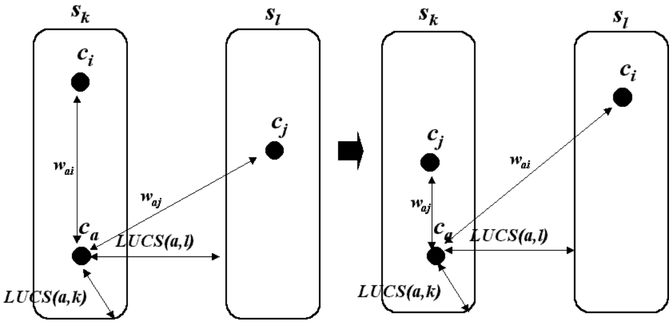

Fig. 8. Cell c_i in switch s_k and c_j in switch s_l are exchanged.

Algorithm: Cell Exchanging

Step 1 For each cell in CG and each switch in G, compute values of matrices $LUCS(i, l)$ and $D(i, l)$. Let $B = C$.
Step 2 Find two cells c_a and c_b assigned to B in different switch such that

$$exchange(a, b) = \max_{(c_i, c_j) \in C \times C} exchange(i, j).$$

Step 3 If $exchange(a, b) > 0$, then exchange cells c_a and c_b; and delete c_a and c_b from B.
Step 4 Update $LUCS(i, l)$, and $D(i, l)$ for each $c_i \in B$ and s_l in L.
Step 5 If B is nonempty go to Step 2.
Step 6 If $exchange(c_a, c_b) > 0$ then go to Step 1; otherwise terminate the algorithm.

6. CELL MIGRATION PHASE

The basic idea of Cell Migration Phase is to assign cells of a used switch to another empty switch in order to reduce the total cost. Without loss of generality, let P_i, $i = 1, ..., m' < m$ be the partition of CG. Cells of partition P_i are assumed to be assigned to switch s_i, $i = 1, ..., m'$. Define $Cabling(k)$ as the sum of the cabling cost associated with switch s_k. That is,

$$Cabling(k) = \sum_{c_i \in P_k} l_{ik}.$$

Let $Cabling(k, l)$ be the reduced cost of all cells after reassigning them from switch s_k to switch s_l, assuming switch s_l is empty, i.e.,

$$Cabling(k, l) = \sum_{i \in P_k} (l_{ik} - l_{il}).$$

Let $Cabling(k, k) = Cabling(k)$ and

$$LUSS(k, l) = \sum_{c_i \in P_k} LUCS(i, k).$$

Lemma 4. If switch s_l is empty and the cells in switch s_k are migrated to switch s_l, then

(1) The reduce location update cost is

$$Mig(k, l) = 2\alpha \sum_{s_z \in S, z \neq k, z \neq l} LUSS(z, k)(d_{zl} - d_{zk}).$$

(2) The reduce total cost is

$$migrate(k, l) = Cabling(k, l) + Mig(k, l).$$

Proof: Consider the migration process shown in Fig. 9; $LUSS(k, z)$ and $LUSS(z, k)$ decrease to 0, and *LUSS*(*z*, *l*) and *LUSS*(*k*, *z*) increase the original value of *LUSS*(*k*, *z*). Thus, it is easy to prove that the reduced location update cost is

$$2\alpha\sum_{s_z \in S, z \neq k, z \neq l}(LUSS(z,k) \times d_{zk} - LUSS(z,l) \times d_{zl}).$$

Since

$$LUSS(z,k) = LUSS(z,l),$$
$$Mig(k,l) = 2\alpha\sum_{s_z \in S, z \neq k, z \neq l}(LUSS(z,k)(d_{zl} - d_{zk}). \qquad \square$$

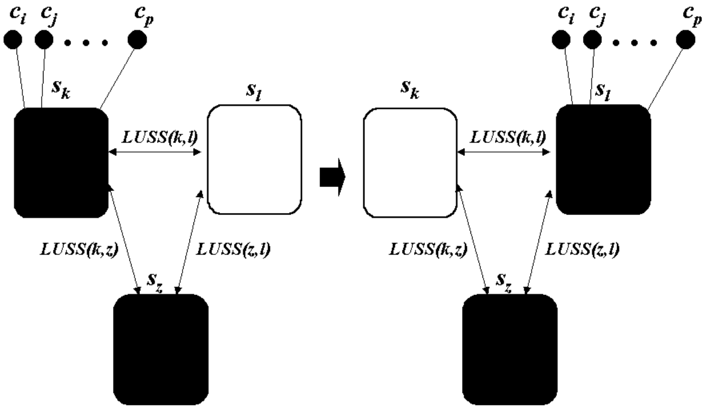

Fig. 9. Cells in switch s_k are migrated to switch s_l.

The following lemmas can be easily derived:

Lemma 5. After migrating cells in switch s_k to switch s_l, *LUSS* can be updated in linear time.

Lemma 6. After migrating cells in switch s_k to switch s_l, *Mig* can be updated in $O(n^2)$ time.

Given an initial assignment, Cell Migration Phase sequentially selects two switches to migrate. At each iteration, a used switch s_a and an empty switch s_b are selected to maximize the reduced migrated cost *migrate*(*a*, *b*), where

$$migrate(a,b) = \max_{s_k \in S, s_l \in S} migrate(k,l).$$

The iteration continuous while *migrate*(*a*, *b*) > 0.

Algorithm: Cell Migration

Step 1 Let a group of switches *S* be represented by *B*. Calculate *LUSS*(*k*, *l*), *Cabling*(*k*, *l*), and *Mig*(*k*, *l*) for switches $s_k, s_l \in S$;

Step 2 Find a used switch s_a and an empty switch s_b in *B* such that

$$migrate(a,b) = \max_{s_k \in S \wedge s_l \in S} migrate(k,l).$$

Step 3 If *migrate*$(a, b) > 0$, migrate cells in switch s_a to switch s_b.
Step 4 Update *LUSS*(k, l), *Mig*(k, l), and *Cbl*(k, l) for s_k, s_l in L.
Step 5 If *migrate*$(a, b) > 0$ then go to Step 2; otherwise terminate.

7. EXPERIMENTAL RESULTS

The proposed algorithm consists of three phases (Cell Pre-Partitioning Phase, Cell-Exchanging Phase, and Cell-Migrating Phase), termed *MCMLCF*(Maximum cell and minimum local communication first), which is a heuristic approach to a rather complex problem. In order to evaluate its performance, we have implemented the algorithm and applied it to a number of examples with randomly positioned cells and switches. The results of these experiments are reported below.

In all the experiments, the implementation language is *C*, and some experiments have been made on Windows NT with a Pentium II 450MHz CPU and 256MB RAM. We simulated a hexagonal system in which cells were configured as an H-mesh. The handoff frequency f_{ij} for each border was generated by a normal random number generator with mean 100 and variance 20. To examine the effects of a different number of cells, Cell Graph *CG* with n = 100, 200, 300, 400, and 500 cells were tested. We used $m = 100$, $\alpha = 1/100$ and $Cap/n = 0.1$ for each problem. The ratio of the number of switches used to the total number of switches is 1/10.

To measure the performance of each phase of the three-phase heuristic MCMLCF algorithm, we simply construct a heuristic algorithm termed *NSF* (Nearest Switch First) as the Cell Pre-partition Phase by assigning a cell to the nearest switch. If the nearest switch is full then find the next nearest. Then, switches are sorted in nondecreasing order according to their load of switches, and run Packing Step to reduce the number of switches being assigned to m'. It is clear that no further assignment process is needed.

For all experiments, the CPU time in seconds of the heuristic algorithm and the objective cost reduction ratio are the major concerns. First, the CPU time for running the three-phase algorithms NSF and MCMLCF by simulating with different number of cells are shown in Table 1. T_{NSF} and T_{MCMLCF} are the CPU times for the problem by running NSF and MCMLCF algorithms in three-phase, respectively. Taking the CPU time for the algorithm NSF as the reference, the CPU time reduction ratio is computed by $(T_{NSF} - T_{MCMLCF})/T_{NSF}$ and shown in Table 1. We found that the CPU time reduction ratio of MCMLCF to NSF is 30.1% on average.

Table 1. CPU time in seconds for algorithm.

# of cells	NSF	MCMLCF	$(T_{NSF} - T_{MCMLCF})/T_{NSF}$
100	14.9	8.8	41.3
200	102.1	68.9	32.5
300	458.1	324.1	29.3
400	1336	1023	23.7
500	3030	2320	23.4
average			30.1

Second, not only is the cost reduction ratio of the whole algorithm evaluated, the cost reduction ratio of the Cell Exchanging Phase, and Cell Migrating Phase are evaluated as well. To test the effects of Cell Exchanging Phase and Cell Migrating Phase, we compare the costs of the algorithm running the two phases. Let C_A, C_B and C_C be the costs resulting from running Cell Pre-Partitioning, Cell Exchanging, and Cell Migrating Phases, respectively. The cost improvement ratios $(C_A - C_B)/C_A$, $(C_B - C_C)/C_B$, and $(C_A - C_C)/C_A$ are shown in Tables 2, 3 and 4, respectively. As seen in Table 4, after running Cell Exchanging and Cell Migrating Phases, the total cost of the NSF algorithm is reduced by 22.1% and the total cost of MCMLCF algorithm is reduced by 34.1%.

Table 2. Reduction in cost ratio of Cell Exchange Phase.

# of cells	NSF	MCMLCF
100	4.2%	6.5%
200	3.2%	9.3%
300	3.7%	9.7%
400	3.1%	7.9%
500	3.3%	19.0%
average	3.5%	11.1%

Table 3. Cost reduction ratio of Cell Migration Phase.

Algorithms	NSF	MCMLCF
100	16.8%	28.8%
200	15.8%	26.4%
300	16.8%	28.3%
400	18.8%	28.1%
500	25.1%	34.6%
average	18.7%	29.2%

Table 4. Cost reduction ratio of Cell Exchange and Cell Migration Phases.

Algorithms	NSF	MCMLCF
100	16.8%	36.8%
200	17.8%	26.8%
300	22.0%	40.8%
400	27.8%	28.4%
500	26.1%	37.6%
average	22.1%	34.1%

The integer programming method failed to find any solution when the problem size grew beyond 30 because the constraint set grew too large for the memory available. For comparison we also implemented a genetic algorithm [16] to solve the same problem. The experimental parameters of the genetic algorithm are: population size is 50, crossover probability is 1.0, mutation probability is 0.05, the number of generations is 1000. The solution found nearly optimal in the objective value. We compare the performance of the NSF and MCMLCF heuristic algorithms against the genetic algorithm. The results in Table 5 show that the heuristic method MCMLCF came close to the optimum - on average within 5%.

Table 5. Comparison of the Heuristic algorithms and GA.

# of cells	NSF/GA	MCMLCF/GA
100	133%	103%
200	122%	110%
300	124%	100%
400	114%	110%
500	117%	103%
average	122%	105%

8. CONCLUSIONS

In this paper, we have investigated the problem of obtaining the optimal assignment of cells in *PCS* (*Personal Communication Service*) to switches in a wireless ATM network. Given cells and switches on an ATM network (whose locations are fixed and known), the problem is to group cells into clusters and to assign these clusters to switches in an optimum manner.

This problem has been modeled as a complex integer programming problem, and the optimal solution of this problem have been found to be *NP-hard*. A three-phase heuristic algorithm, MCMLCF, consisting of *Cell Pre-Partitioning Phase, Cell Exchanging Phase*, and *Cell Migrating Phase* is proposed. First, in the Cell Pre-partitioning phase, a three-step procedure (*Clustering Step, Packing Step, and Assigning Step*) is proposed to group cells into clusters. Second, Cell Exchanging Phase is proposed to improve the result by repeatedly exchanging two cells in different switches so as to provide the greatest improvement. Finally, Cell Migrating Phase is proposed to reduce cost by repeatedly migrating all cells in a used switch to an empty switch.

Further, we evaluate the performance of the Cell Exchanging Phase, Cell Migrating Phase, and the whole algorithm (MCMLCF). Experimental results indicate that the proposed algorithm runs efficiently. Comparing the results of the algorithm to the NSF algorithm, the computation time is reduced by 30.1%. The results show that Cell Exchanging Phase and Cell Migrating Phase reduce the total cost as well, and by comparison to the result of the genetic algorithm, the heuristic method MCMLCF came close to the optimum -on average within 5%. The computation time of algorithm MCMLCF to NSF is reduced by 30.1%.

ACKNOWLEDGMENT

This work was supported in part by the MOE Program of Excellence Research under Grant 89-E-FA-04-1-4.

REFERENCES

1. M. Cheng, S. Rajagopalan, L. F. Chang, G. P. Pollini, and M. Barton, "PCS mobility support over fixed ATM networks," *IEEE Communication Magazine*, Vol. 18, No. 2, 1997, pp. 82-91.
2. D. C. Cox, "Personal communication–a viewpoint," *IEEE Communication Magazine*, Vol. 11, No. 8, 1990, pp. 8-20.
3. U. Varshney, "Internet watch: supporting mobile using wireless ATM," *IEEE Computer*, Vol. 30, No. 1, 1997, pp. 131-133.
4. T. Hattori, "Personal communication-concept and architecture," in *International Conference on Communications* (*ICC*'90), 1990, pp. 333.4.1-335.4.7.
5. R. Jain, Y. B. Lin, C. Lo, and S. Mohan, "A caching strategy to reduce network impacts of PCS," *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 8, 1994, pp. 1434-1444.
6. Y. B. Lin, "Determining the user locations for personal communications services networks," *IEEE Transactions on Vehicular Technology*, Vol. 43, No. 3, 1994, pp. 466-473.
7. Y. B. Lin and S. Y. Hwang, "Comparing the PCS location tracking strategies," *IEEE Transactions on Vehicular Technology*, Vol. 45, No. 2, 1996, pp. 114-121.
8. S. Mohan and R. Jain, "Two user location strategies for personal communications services," *IEEE Personal Communications*, Vol. 1, No. 1, 1994, pp. 42-50.
9. D. Raychaudhuri and N. Wilson, "ATM based transport architecture for multiservices wireless personal communication network," *IEEE Journal on Selected Areas in Communicaions*, Vol. 12, No. 8, 1994, pp. 1401-1403.
10. A. Merchant and B. Sengupta, "Assignment of cells to switches in PCS networks," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 5, 1995, pp. 521-526.
11. R. Steele, "Deploying personal communication networks," *IEEE Communication Magazine*, Vol. 4, No. 1, 1990, pp. 6-11.
12. M. D. Yacoub, *Foundations of Mobile Radio Engineering*, CRC Press, Boca Raton, FL, 1993.
13. E. L. Laeler, *Combinatorial Optimization: Networks and Matroids*, Rinehart & Winston, New York, Holt, 1976.
14. H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, Vol. 2, No. 1, 1955, pp. 83-97.
15. B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Systems Technical Journal*, Vol. 49, No. 3, 1970, pp. 291-307.
16. D. R. Din and S. S. Tseng, "Genetic algorithms for optimal design of two-level wireless ATM network," to appear in *Proceeding of NSC*.

Der-Rong Din (丁德榮) received the M.S. degree in Computer and Information Science from National Chiao Tung University, Taiwan, R.O.C., in 1994, and is currently a doctoral candidate at National Chiao Tung University. His research interests include wireless ATM, parallel processing, computer algorithm, and parallelizing compiler design.

Shian-Shyong Tseng (曾憲雄) received his Ph.D. degree in Computer Engineering from the National Chiao Tung University in 1984. Since August 1983, he has been on the faculty of the Department of Computer and Information Science at National Chiao Tung University, and is currently a Professor there. From 1988 to 1991, he was the Director of the Computer Center at National Chiao Tung University. From 1991 to 1992 and 1996 to 1998, he acted as the Chairman of Department of Computer and Information Science. Form 1992 to 1996, he was the Director of the Computer Center at Ministry of Education and the Chairman of Taiwan Academic Network (TANet) management committee. In December 1999, he founded Taiwan Network Information Center (TWNIC) and is now the Chairman of the board of directors of TWNIC. His currently research interests include parallel processing, expert systems, computer algorithm, and internet-based application.