MSP-CACRR: Multidimensional Sequential Patterns Based Call Admission Control and Resource Reservation for Next-Generation Wireless Cellular Networks

Sherif Rashad
Mathematics and Computer Science Department
Morehead State University, Morehead, KY 40351, USA
s.rashad@moreheadstate.edu
Phone: (606) 783-9179  Fax: (606) 783-5002

Mehmed Kantardzic and Anup Kumar
Computer Engineering and Computer Science Department, University of Louisville
Louisville, KY 40292, USA
{mmkant01, ak}@louisville.edu

Abstract—In this paper, we propose and evaluate a multidimensional sequential pattern (MSP) based CAC and RR technique called MSP-CACRR technique which is designed for next generation of cellular networks. MSP-CACRR is based on an efficient multidimensional sequence mining approach called MobilePrefixSpan and is designed for the wireless cellular networks. MobilePrefixSpan is an extension version of the well-known sequence mining technique PrefixSpan and its multidimensional sequences version. The spatial, temporal, and usage information are used in MobilePrefixSpan to extract the movement patterns of mobile users. Simulation was conducted to study the effect of each of these dimensions on the network performance. Simulation results show that the proposed MSP-CACRR technique uses the mobility patterns effectively to enhance the performance of cellular networks.

I. INTRODUCTION

The recent revolution of mobile communications have enabled a new generation of applications that require a broad range of radio resources (bandwidths). The scarcity of the available bandwidth and the mobility of the users make the problem of radio resource management (RRM) more challenging in the wireless networks [1, 2]. Call admission control (CAC) and resource reservation (RR) are critical activities in radio resource management. CAC deals with the task of deciding if a new or handoff call should be admitted into and supported by the network [1]. A call can be considered as a handoff call if the user is moving while this call is still in progress. To ensure a guaranteed quality of service (QoS), the service providers have to maintain the ongoing calls when the mobile users move from one cell to another cell. Moreover, a cell has to accept as many new calls as possible taking into account the handoff calls will always have a priority over the newly arriving calls. Because of user’s mobility, resource reservation is used to maintain an ongoing call without dropping.

Call blocking probability (CBP), call dropping probability (CDP), and bandwidth utilization (BWU) are the important parameters to evaluate the performance of the CAC and RR techniques. When there is not enough bandwidth to serve the handoff calls, this call may be dropped. In addition, if there is not enough bandwidth to serve the new arriving call, these calls may be rejected (blocked). Bandwidth utilization (BWU) is an important parameter that is used to measure the effective bandwidth usage. It is defined as the ratio between the amount of bandwidth used by various applications admitted into the network and either the total requested bandwidth or the total available bandwidth, whichever is the smaller [1].

The user mobility prediction and estimation have been used to enhance the performance of the call admission control, resource reservation, location management and handover management in mobile networks. We believe that using detailed temporal information will enhance prediction. We have introduced in [3] the PCAC-RR technique that used the spatial information (location of mobile users) and temporal information (the time of the day). It has been shown that the PCAC-RR has a better performance compared to other techniques. The PCAC-RR was based on a simple one-dimensional sequence mining techniques to extract the mobility profiles. The main limitation of this approach is that it does not consider the frequent subsequences of cells to predict the new location. This will limit the ability to predict the correct location of the mobile users in many situations. Also, it assumes similar performance for the mobile users during the weekdays and during the weekends. This assumption may not be valid in many cases because the behavior of many mobile users will be different during the weekends [4].

In this paper, we propose and evaluate a multidimensional sequential patterns (MSP) based CAC and RR technique called MSP-CACRR. This technique is based on using the spatial information and the temporal information of the mobile users. We have developed a data mining technique called MobilePrefixSpan that can work in the mobile environment and uses the collected mobility information to generate the frequent paths of the mobile users. MobilePrefixSpan is a modification of the well-known PrefixSpan technique and its multidimensional sequence mining version. We also evaluate the importance of using the temporal information in addition to the spatial information to improve the prediction accuracy.

In the next section, we explain the MobilePrefixSpan technique in details. Sections three and four explain in details the proposed CAC and RR techniques. The experimental results are discussed in section 5. Finally, we give our conclusion in section 6.
II. PROPOSED METHODOLOGY: MSP-CACRR

The proposed MSP-CACRR scheme is mainly based on generating two types of users’ mobility profiles. The first type is called the local mobility profile. This profile is generated individually for each user by his/her mobile host (MH). The second type of the mobility profiles is called the global mobility profile generated by each base station (BS). The global profile will be used if the local profile gives low accuracy for future mobility prediction.

There are two types of information that are collected and used to generate the mobility patterns:

- Spatial Information: this indicates the location of the mobile user at every recording interval. This information is collected by recording the ID of each base station (BS).
- Temporal Information: this indicates the time and the day information collected during the navigation of the mobile users. At each recorded movement we register also the day and the time at which this movement was performed.

A. Data Collected by the Mobile Host (MH)

The mobility data items required for a user are shown on Table 1. These data are collected by the MH of its user.

<table>
<thead>
<tr>
<th>ID of Cell</th>
<th>Start Time</th>
<th>End Time</th>
<th>Weekend</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_1</td>
<td>VST_1</td>
<td>VET_1</td>
<td>W_1</td>
</tr>
<tr>
<td>C_2</td>
<td>VST_2</td>
<td>VET_2</td>
<td>W_2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>C_n</td>
<td>VST_n</td>
<td>VET_n</td>
<td>W_n</td>
</tr>
</tbody>
</table>

where:
- C_i: represents the ID of the current visited cell.
- VST_i: represents the time stamp when this MH enters C_i.
- VET_i: represents the time stamp when this MH exits C_i.
- W_i: a binary value describes if the information was recorded during the weekend or not. We will use the letter Y if it was on a weekend day, otherwise we use the letter N (this means W_i ∈ {Y, N}).

B. Data Collected by the Base Station (BS)

The base station in each cell is responsible for collecting the data that represents the movements of its mobile users. The data items required to be collected for the base station in cell C_i are as shown below in Table 2. BS mobility model will mainly help the mobile users to navigate smoothly from one BS to another BS in case mobility profiles at an MH can not help in prediction.

<table>
<thead>
<tr>
<th>Previous Cell</th>
<th>Start Time</th>
<th>Next Cell</th>
<th>End Time</th>
<th>Weekend</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP_i</td>
<td>VST_i</td>
<td>CN_i</td>
<td>VET_i</td>
<td>W_1</td>
</tr>
<tr>
<td>CP_2</td>
<td>VST_2</td>
<td>CN_2</td>
<td>VET_2</td>
<td>W_2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>CP_n</td>
<td>VST_n</td>
<td>CN_n</td>
<td>VET_n</td>
<td>W_n</td>
</tr>
</tbody>
</table>

where:
- CP_i: represents the ID of the previous visited cell.
- CN_i: represents the ID of the next visited cell and for any record in the table: CP_i ≠ CN_i ≠ C_k

C. Generating Local and Global Paths

We introduce here the hypothesis that user behaviors (and therefore user profiles) are not the same for different periods of the day. Therefore, we propose dividing the day into several intervals. These intervals are determined for the users according to the known rush hours, general work hours, etc. For example, we can divide the day into the following six intervals: T1 (morning rush hours: [6:00 - 9:00]), T2 (morning-noon: [9:00-12:00]), T3 (afternoon [12:00 - 5:00]), T4 (evening rush hour: [5:00-6:00]), T5 (evening: [6:00 - 12:00]), and T6 (after midnight: [12:00 - 6:00]). To generate the movement paths, it is necessary to transform the collected data into a sequence of symbols, where each symbol represents a base station. Every path is composed of a sequence of base stations IDs for each recorded visit. In addition, each path also contains the information regarding the duration of MH stay in a cell. This is achieved by collecting mobility data of a MH at fixed time slots Δt. The basic process of generating the movement paths are explained in details in [3].

D. Multidimensional Sequential Mining for Mobility Patterns: MobilePrefixSpan

In this section we explain the details of the data mining technique, called MobilePrefixSpan, which we have developed to analyze the information collected from the mobile users. MobilePrefixSpan technique is a modified version of the well-known sequence mining technique PrefixSpan [5] and its multidimensional sequences version UNISEQ [6]. These are not applicable in the mobile environment because we need only to consider the subsequence of cells if they are in consecutive order. This problem is solved in the MobilePrefixSpan by introducing what is called consecutive-subsequences (this will be explained in details in next parts of this paper). The goal of MobilePrefixSpan technique is to extract the movement patterns of the mobile users using the collected information. Fig. 1 shows the block diagram that describes the inputs and outputs for the MobilePrefixSpan technique.

1) Problem Definition of Sequence Mining in the Mobile Network Environment

Data mining extracts implicit, previously unknown, and potentially useful information from datasets. Many approaches have been proposed to extract information. One of the most important ones is mining sequential patterns. The sequential pattern mining problem was first introduced by Agrawal and Srikant in [7].

![Fig. 1. Inputs and outputs for MobilePrefixSpan technique.](image-url)
In this section, we give the basic definitions to formalize the problem of sequence mining in the mobile environment. These definitions for MobilePrefixSpan approach are based on the definitions that have been introduced in [5, 6, 8]:

**Definition 1:** Let \( I = \{i_1, i_2, \ldots, i_k\} \) be a set of items such that each item here represents a base station ID (or cell ID). An itemset \( X \) is a subset of items, i.e., \( X \subseteq B \).

**Definition 2:** A sequence is an ordered list of itemsets. A sequence \( S \) is denoted by \( \langle s_1, s_2, \ldots, s_k \rangle \) where \( s_j \subseteq I \) for \( 1 \leq j \leq k \).

In our case each sequence represents an ordered set of base station (cells) IDs. For example, the sequence \( S_1 = \langle C_1, C_2, C_3 \rangle \) represents the transition of the mobile user from cell \( C_1 \) to \( C_2 \) and then to \( C_3 \).

**Definition 3:** The number of instances of items in a sequence is called the length of the sequence.

For example, a sequence with length \( k \) is called a \( k \)-sequence. For example the sequence \( S_1 = \langle C_1, C_2, C_3, C_4, C_5 \rangle \) is a 5-sequence.

**Definition 4:** A sequence \( \alpha = \langle a_1, a_2, \ldots, a_n \rangle \) is called a consecutive-subsequence of another sequence \( \beta = \langle b_1, b_2, \ldots, b_m \rangle \) and denoted as \( \alpha \subseteq \beta \) if all of the ordered itemsets in \( \alpha \) appear in the exact same consecutive order in \( \beta \).

This definition is needed for sequence mining in the mobile environment. Our problem does not only consider the order of visited cells, but also considers the path which contains consecutively ordered cells that are required to move from one cell to another. For example, the 5-sequence \( S_1 = \langle C_1, C_2, C_3, C_4, C_5 \rangle \) has only the following consecutive-subsequences of length 3:

- \( \langle C_1, C_2, C_3 \rangle \), \( \langle C_3, C_4, C_5 \rangle \), and \( \langle C_4, C_5 \rangle \)

Here we didn’t consider other subsequences which violate the consecutive ordering condition such as the subsequence \( \langle C_1, C_2, C_4 \rangle \) where \( C_4 \) in our example never comes directly after \( C_3 \).

**Definition 5:** A sequence dataset \( S \) is a set of tuples \( \langle sid, s \rangle \), where \( sid \) is an identification of the sequence and \( s \) is a sequence.

**Definition 6:** A tuple \( \langle sid, s \rangle \) is said to contain a sequence \( \alpha \), if \( \alpha \) is a subsequence of \( s \) (i.e. \( \alpha \subseteq s \)).

**Definition 7:** The support of a sequence \( \alpha \) in the sequence dataset \( S \) is the number of tuples that contain \( \alpha \), i.e.:

\[
\text{sup}(\alpha) = \left| \left\{ (\langle sid, s \rangle) : \langle sid, s \rangle \in S \land (\alpha \subseteq s) \right\} \right| (1)
\]

**Definition 8:** A multidimensional sequence \( S \) is an ordered set of items (cells) that have common attributes and is represented in the form:

\[
S = \{d_1, d_2, \ldots, d_n, <s_1, s_2, \ldots, s_k>\}
\]

Where

- \( d_i \) is the value of the attribute at the \( i \)th dimension.
- \( n \) is the number of dimensions (attributes)
- \( s_j \) is the base stations IDs.

For example, we have the following two dimensions are used to build the mobility patterns of the mobile users:

- \( d_1 \) : the period of the day. If we have four periods, then \( d_1 \in \{T_1, T_2, T_3, T_4\} \).  
- \( d_2 \) : the type of the day which can be a weekend (Y) or a working day (N), where \( d_2 \in \{Y, N\} \).

In this case, the multidimensional sequence \( S = \{T_2, N, <C_1, C_4, C_5>\} \) represents the transition of the mobile user from cell \( C_1 \) to \( C_4 \) and then to \( C_5 \) at the second period of the day (T2) and during a working day (N).

**E. MobilePrefixSpan Algorithm**

The MobilePrefixSpan algorithm is shown in Fig. 2. To explain the details of the MobilePrefixSpan algorithm, let us assume that we have the multidimensional dataset shown in Table 3 that contains the generated sequences of one of the mobile users. As we can see in this example, there are two dimensions as defined in the previous sections. The first step is to reconfigure the dataset such that the multidimensional information is embedded in the sequences. The new sequences will be as shown in Table 4. This converts the problem to one-dimensional sequence mining problem. Here the first three elements in any of these sequences always refer to the multidimensional information. Once we have these new sequences we can start applying the MobilePrefixSpan technique where we do not consider a subsequence if all the items that represent these cells are not found in the consecutive order (as explained in Definition 4). This rule is not applied to the items that represent multidimensional information. This is because we can combine any of these dimensions together without including the remaining dimensions, while we can not ignore the in-between cells.

The steps of the MobilePrefixSpan algorithm shown in Fig. 2 are explained below with the help of an example using the converted the sequences shown in Table 4. We will use in this example a minimum support value equal to 2 (\( \min_{\text{sup}} = 2 \)) which represents the minimum number of repetitions for the sequence in the given database in order to be considered as a frequent sequence.

1. Prepare the multidimensional sequences and convert the problem to one-dimensional sequence mining.
2. Find the projected dataset corresponding to the single-item frequent sequences.
3. Find the frequent sequences using this prefix.
4. If the items represent BSs, then consider the consecutive order.
5. Record the frequent sequences that have been found using this prefix.
6. Use each of these recorded frequent sequences as a prefix to find its projected dataset.
7. Repeat steps from 4 to 7 until we find all sequential patterns.
MobilePrefiSpan will be applied as follows:

1. Apply the first scan to find all of the single-item frequent sequences (prefix).
   - In our example, these frequent sequences are as shown in Table 5 with the corresponding supports.
2. Find the projected datasets corresponding to the single-item frequent sequences.
   - The projected dataset consists of postfix sequences which contains all the frequent items that follow the first occurrence of the prefix at any sequence.
   - For example, if we consider the prefix < T1 > in our example, we will find that the projected dataset contains the following two postfix sequences: < Y Class-I C2 C3 C4 C5 > and < Y C6 C2 C3 C4 >.
3. We continue by finding out the single-item frequent sequences in each projected dataset. In our example these will be < Y >, < C3 >, < C1 >, and < C4 >.
4. Record the frequent sequences that have been found using this prefix and previous ones. Now, we have four two-item frequent sequences in our example which are < T1 Y >, < T1 C2 >, < T1 C3 >, and < T1 C4 >.
5. Use each of these recorded frequent sequences as a prefix to find its projected dataset. For example, the projected dataset of the prefix < T1 C2 > contains the following two postfix sequences: < C3 C4 > and < C1 C4 >. In this case we will find that the single-item frequent sequences in this projected database are < C3 >, and < C4 >. Here, we can consider only one three-item frequent sequence which is < T1 C2 C3 > but we can’t consider < T1 C2 C4 > because C4 doesn’t follow C2 directly in this case but it has to be through C1.
6. Repeat steps from 3 to 5. If we continue to find the projected database of the prefix < T1 C2 C3 > we will find that we have < C3 C4 > and < C4 >. Here, < C4 > is the only single-item frequent sequence and this generates the four-item frequent sequence < T1 C2 C3 C4 >. After that we will not find more frequent sequences using the prefix < T1 C2 C3 > because the projected database will have only the postfix < C4 >, and it does not satisfy the minimum support. At this point we can say that we have found all of the frequent sequences (or sequential patterns) using the prefix < T1 C2 >.
7. This process will continue through the database for all of the items until we find all of the sequential patterns.

For the application of our problem, we will use only the frequent sequences that contain at least two cells IDs. For example, we can not use the frequent sequence < T1 Y C3 > because it contains only the information that C3 is frequent in the first period and during the weekend, but it does not have any information about the transition from one cell to another which is the most important information that we are trying to discover.

Also, we introduce and apply what we will call last-item rule to simplify the algorithm. The last-item rule states that “when we calculate the support to any item there is no need to consider the sequences where that item appears as the last element in these sequences”. This is because this item will not have any postfix in these sequences. This will automatically eliminates some items form the set of frequent items. For example, without applying the last-item rule we have < C3 > with support of 2 as a frequent item as show in Table 5. We can see that in Table 6 we will not find any frequent sequences using < C3 > as a prefix. By applying the rule, these results can be deduced form the beginning when we recalculate the support of < C3 > to be 1 (without including the first sequence in Table 4) and therefore it will not be a frequent item. The stored datasets of all possible sequential patterns will be as shown in Table 6. As we can see in this database, we consider only the paths with at least two cells IDs.

<table>
<thead>
<tr>
<th>sid</th>
<th>Time Int.</th>
<th>Weekend</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T1</td>
<td>Y</td>
<td>&lt; C1, C3, C4, C5 &gt;</td>
</tr>
<tr>
<td>2</td>
<td>T2</td>
<td>Y</td>
<td>&lt; C6, C1, C4 &gt;</td>
</tr>
<tr>
<td>3</td>
<td>T1</td>
<td>Y</td>
<td>&lt; C1, C2, C3, C4 &gt;</td>
</tr>
<tr>
<td>4</td>
<td>T2</td>
<td>N</td>
<td>&lt; C3, C4, C5 &gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sid</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt; T1, Y, C1, C2, C3, C4, C5 &gt;</td>
</tr>
<tr>
<td>2</td>
<td>&lt; T2, Y, C6, C1, C4 &gt;</td>
</tr>
<tr>
<td>3</td>
<td>&lt; T1, Y, C6, C2, C3, C4 &gt;</td>
</tr>
<tr>
<td>4</td>
<td>&lt; T2, N, C3, C7, C4, C5 &gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Single-item frequent sequences</th>
<th>Support</th>
<th>Single-item frequent sequences</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; T1 &gt;</td>
<td>2</td>
<td>&lt; C1 &gt;</td>
<td>4</td>
</tr>
<tr>
<td>&lt; T2 &gt;</td>
<td>2</td>
<td>&lt; C1 &gt;</td>
<td>2</td>
</tr>
<tr>
<td>&lt; Y &gt;</td>
<td>3</td>
<td>&lt; C4 &gt;</td>
<td>2</td>
</tr>
<tr>
<td>&lt; C1 &gt;</td>
<td>2</td>
<td>&lt; C3 &gt;</td>
<td>3</td>
</tr>
</tbody>
</table>

F. Calculating the Support of Sequential Patterns

As we can see in Table 6, we have calculated two types of supports (the general support and the projected support) for each frequent sequence. The general support refers to the support of the sequence in the database and it is defined by: \( \text{supDB}(S_i) \) is the support of the sequence \( S_i = < s_1, s_2 \ldots s_k > \) in the dataset DB.

For example, if we want to calculate the support of the frequent multidimensional sequence \( S_i = < C_6, C_3 > \) shown in Table 6, we will find that \( S_i \) is appearing in 3 rows of the dataset shown in Table 3 (where the total number of records
Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

...in the dataset is 4), therefore the support of S₁ is equal to 0.75 (we write it as sup_D(S₁)=0.75). This is very helpful because it gives indication about the repetition of the sequence in all the cases.

The projected support refers to the support of the sequence at the projected dataset according to the required dimensions and it is defined by:

\[ \text{sup}_D(S_i, D_j) = \text{projected support of the sequence } S_i \text{ at projected dataset from DB given by the dimensions } D_j. \]

The projected support of a frequent multidimensional sequence \( S_i^m < s_1, s_2, \ldots, s_k > \) with dimensions \( D_j^m = \{ d_1, d_2 \ldots, d_m \} \) is calculated by looking to the projected database that have the same dimensions \( \{ d_1, d_2 \ldots, d_m \} \) and find the ratio of the records that have the sequence \( s_1, s_2, \ldots, s_k > \) in this projected database. For example, if we look at the frequent multidimensional sequence \( S_i^m < C_6, C_7 > \) with \( D_j^m = \{ * Y \} \) as shown in Table 6, we will find that there are three records in the original datasets that have the same dimensions \( * Y \) and two of them have the sequence \( C_6, C_7 > \). Therefore we can say that the projected support is 0.667 (we write \( \text{sup}_D(S_i, D_j) = 0.667 \)). This type of support is an important measure since it gives us information about how the frequent sequence was repeated within the required dimensions compared to other sequences that have the same dimensions.

These two types of supports will be used as the main parameters to determine the effectiveness of the sequence. The resources that will be reserved for each user for their future movements will be based on this effectiveness value. For each sequence \( S_i^m < s_1, s_2, \ldots, s_k > \), we use the two supports to generate the effectiveness by using the dimension of the sequence set given the set of \( n \) dimensions \( D_j = \{ d_1, d_2, \ldots, d_m \} \), and this effectiveness is denoted as \( E(S_i / D) \) which can be calculated as:

\[
E(S_i | D_j) = \text{sup}_D(S_i, D_j) \cdot w_1 \cdot w_2
\]

and

\[
w_1 = \begin{cases} \frac{\text{sup}}{\text{sup}_D(S_i, D_j)} & \text{if } \text{sup} > \text{sup}_D(S_i, D_j) \\ 1 & \text{if } \text{sup} \leq \text{sup}_D(S_i, D_j) \end{cases}
\]

\[
w_2 = \begin{cases} 1 & \text{if } \text{sup} D(S_i, D_j) \geq \text{sup}_D(S_i, D_j) \\ \frac{\text{sup}_D(S_i, D_j)}{\text{Max}_{sup}} & \text{if } \text{sup} D(S_i, D_j) < \text{sup}_D(S_i, D_j) \end{cases}
\]

\[\text{Max}_{sup} = \max_{i=1}^{n} \left( \max_{s} (\text{sup}_D(S_i, D_j)) \right)\]

where \( \text{Max}_{sup} \) is the maximum projected support found in all of the discovered sequences, \( N_D \) is the number of possible combinations between the dimensions, \( N_i \) is the total number of frequent sequences.

Our model of frequent sequences is modified into the model of effective sequences, where the effectiveness of a sequence reflects the power of that sequence in the DB compared to the other sequences. This was accomplished by considering the weights \( w_1 \) and \( w_2 \) shown in equation 2. The factor \( w_1 \) was used to ensure that we consider the support in the whole database if it is larger than the support in the projected dataset. On the other hand the factor \( w_2 \) was used to ensure that we consider support in the projected dataset if it is larger than the support in whole database and it also gives information about the effectiveness of this sequence compared to most frequent sequence in all projected databases.

The resources that will be reserved for each user for their future movements will be based on this effectiveness value \( E(S_i / D) \). If there is no effect of considering the dimensions we will use the general support \( \text{sup}_D(S_i) \) for resource reservation. Otherwise we will use the values of the projected support which has been modified by a ratio of this support to the maximum support.

**Table 6: Stored Sequential Patterns and Corresponding Support and Confidence**

<table>
<thead>
<tr>
<th>Time Interval</th>
<th>Confid.</th>
<th>Frequent Sequence</th>
<th>Support D</th>
<th>Support proj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>T0</td>
<td>0</td>
<td>(&lt;C_6 /&gt;)</td>
<td>2/4=0.5</td>
<td>2/4=0.5</td>
</tr>
<tr>
<td>T0</td>
<td>1</td>
<td>(&lt;C_7 /&gt;)</td>
<td>2/4=0.5</td>
<td>2/4=0.5</td>
</tr>
<tr>
<td>T1</td>
<td>2</td>
<td>(&lt;C_6 /&gt;)</td>
<td>2/4=0.5</td>
<td>2/4=0.5</td>
</tr>
<tr>
<td>T1</td>
<td>3</td>
<td>(&lt;C_6 /&gt;)</td>
<td>2/4=0.5</td>
<td>2/4=0.5</td>
</tr>
<tr>
<td>T2</td>
<td>4</td>
<td>(&lt;C_6 /&gt;)</td>
<td>3/4=0.75</td>
<td>2/4=0.5</td>
</tr>
<tr>
<td>T2</td>
<td>5</td>
<td>(&lt;C_6 /&gt;)</td>
<td>3/4=0.75</td>
<td>2/4=0.5</td>
</tr>
<tr>
<td>T2</td>
<td>6</td>
<td>(&lt;C_6 /&gt;)</td>
<td>3/4=0.75</td>
<td>2/4=0.5</td>
</tr>
</tbody>
</table>
The service providers determine the total bandwidth assigned to each cell during the design phase. The bandwidth used by the users in cell \( C_i \) at the expected handoff time \( T_h \) can be estimated easily from the bandwidth used at the current time. If there is a call, the required amounts of bandwidth to be reserved in the other cells have to be calculated to handle the handoff operation. The expected time at which the handoff will happen has to be taken into account as well. Suppose that there is a call in cell \( C_i \) at time \( T_h \). The proposed algorithm of the RR scheme will be executed as follows:

1. Determine the corresponding dimensions form the current time and current day. In this case we can determine \( D_k = \{ d_1, d_2 \} \) where \( d_1 \) refers to the time interval and \( d_2 \) refers to the type of the day.
2. According to the set of dimensions \( D_k \), start to look at the sequential patterns stored in the MH of the current user that are corresponding to these dimensions to predict the frequent sequences.
3. If all of the current dimensions can not be supported by the stored sequential patterns, we use \( D_k = \{ *, * \} \) to predict the frequent sequences.
4. If there is at least one predicted frequent sequence, then the bandwidth reservation is performed based on sequential patterns stored in the MH as follows:

   i. For each cell \( C_k \) in the predict sequence \( (S_i) \), estimate the handoff time \( (T_h) \) using the known time slot \( \Delta t \).
   ii. Once we find \( T_h \), check the available bandwidth at \( T_h \). The available bandwidth could be calculated from equation (6).
   iii. The required amount of bandwidth \( B_{req} \) to be reserved is determined as a portion from the actual required bandwidth \( B_{actual} \) according to the effectiveness of the path using the following equation:

\[
B_{req} = B_{actual} \times E(S_i,D_k) \tag{7}
\]

   iv. If there is enough bandwidth available at time \( T_h \), then the amount of bandwidth to be reserved will be \( B_{req} \) and then the available and the reserved bandwidths, are updated.
   v. If there is not sufficient bandwidth available and the call is from Class-I, then this call can borrow some amount \( (B_{borrow}) \) form the Class-II calls in that cell.
   vi. Otherwise the amount of bandwidth will be the available bandwidth plus \( B_{borrow} \).

5. If no frequent sequence could be predicted using sequential patterns stored in the MH, then we have to access the sequential patterns stored in the BS, and use the frequent sequences in a way similar to the one described in the step 2 to reserve the required bandwidth.

IV. PREDICTIVE CALL ADMISSION CONTROL SCHEME

The admitted call may be a new call or a handoff call. Each of these categories of calls will be handled in a different way taking into account that the handoff calls will have a priority over the new calls. If the admission is performed at the cell \( C_i \) and at the time interval \( T_j \), then the CAC algorithm can be given as follows:

a. For the new calls from Class-I:

   i. According to the set of dimensions \( D_k \), start to look at the sequential patterns stored in the MH of the current user that are corresponding to these dimensions to predict the frequent sequences.
   ii. If all of the current dimensions can not be supported by the stored sequential patterns, we use \( D_k = \{ *, * \} \) to predict the frequent sequences. If no frequent sequence could be predicted using sequential patterns stored in the MH, then we have to access the sequential patterns stored in the BS.
   iii. According to \( D_k \) and the current cell \( C_i \), check the available BW in the current cell \( C_i \) (which is given by \( B_{req}(C_i,T_j) \) as well as BW in the next expected calls in the predicted frequent sequence with high probabilities above a predetermined threshold \( L_p \). In our case we used \( L_p=0.8 \).
   iv. If each of the next expected calls has available BW to cover this new call at the next time step, then the call is accepted and the required bandwidth is allocated.
   v. Otherwise the call is blocked.

b. For the new calls from Class-II:

   i. Check only the available BW at the current cell \( C_i \), at which the admission is performed.
   ii. If the required BW is less than equal the \( B_{req}(C_i,T_j) \), then the call is accepted and the required BW is allocated.
   iii. If the required BW is greater than \( B_{req}(C_i,T_j) \), and there is any available BW \( B_{req}(C_i,T_j) \neq 0 \), then the call is accepted and the available BW is allocated. Then, this bandwidth readjusted in future when there is BW available.
   iv. Otherwise the call is blocked.

c. For the handoff calls from Class-I:

   i. Handoff calls can allocate bandwidth not only from the reserved bandwidth, but also from the available bandwidth (while the new calls can allocate bandwidth from the available bandwidth only).
   ii. Check the bandwidth reserved for the handoff calls added to the available bandwidth in the cells at that time interval \( B_{req}(C_i,T_j) + B_{req}(C_i,T_j) \).
   iii. If the required bandwidth \( \leq B_{req}(C_i,T_j) + B_{req}(C_i,T_j) \), then the call is accepted and the bandwidth is allocated according to the required bandwidth.
   iv. Otherwise the call is dropped.

d. For the handoff calls from Class-II:

   i. The call will be accepted if there is bandwidth available in the reserved or the available bandwidths.
ii. If the required BW is less than or equal the \([B_i(C_i, T_j)]\), then the call is accepted and the required BW is allocated.

iii. If the required BW is greater than \([B_i(C_i, T_j)]\) and \([B_i(C_i, T_j)] \neq 0\), then the call is accepted and the allocated BW will be the minimum of \([B_i(C_i, T_j)]\) and \([B_i(C_i, T_j)]\). Then, this bandwidth readjusted in future when there is BW available.

iv. Otherwise the call is blocked.

This approach will maximize the probability of completing the call without dropping during the user navigation through the cells.

V. EXPERIMENTAL RESULTS

The simulation was conducted to evaluate the performance of the proposed MSP-CACRR technique and compare its performance with other predictive techniques. We compare the performance of the MSP-CACRR with three other schemes: PCAC-RR scheme (described in [3]), PR-CAT4 scheme (described in [9]) and the FR-CAT2 scheme (described in [9] and [10]). We used the same topology model and the same simulation parameters that have been used in [3]. The call arrival rate is assumed to follow a Poisson distribution with mean \(\lambda\) (calls/second) in each cell (from 0.01 to 0.1). The call duration (call service time) is assumed to be exponentially distributed with mean of 180 seconds.

It has been demonstrated in [3] that the best number of time intervals that can be used for the users of the simulated mobile networks described in [3] will be four time periods. Therefore, in all of the following experiments, we will use these four time intervals of the day: T1: [6:00 am - 12:00 pm], T2: [12:00 pm - 6:00 pm], T3: [6:00 pm - 12:00 am], and T4: [12:00 pm - 6:00 am].

A. Study the Effect of Using the Time interval and the Type of the Day as Two Dimensions

In order to study the effect of adding the type of the day (weekend or working day) as one of the dimensions, we conducted the simulation to study the performance of the MSP-CACRR technique using two dimensions: the time intervals of the day and the type of the day. We refer to this technique as MSP-CACRR(d1, d2). The overall performance results are shown in Fig. 3 (a,b, and c). As we can see from these results, using the type of the day as one of the dimensions enhances significantly the performance of the MSP-CACRR technique. This supports the hypothesis that the behavior of most of the mobile users will not be same during the weekend days. If we did not differentiate between the weekends and the working days, as in the MSP-CACRR(d1), the prediction of the frequent sequences will not be accurate in most of the cases during the weekend days. In this case, the network will not be able to utilize the provided BW effectively. Hence, it is very important to add the type of the day (weekend or working day) as one of the dimensions. This dimension will be used also in the next phase of comparisons. With very high loads, most of the resources will be reserved in both cases to complete the calls and this why we see that the BWU will be with high rates in both cases.

![Graphs](Fig. 3 Comparing between MSP-CACRR(d1) and MSP-CACRR(d1, d2)

B. Performance of MSP-CACRR Technique Compared to Other Techniques

The simulation was also conducted to compare performance of the MSP-CACRR technique with PCAC-RR, PR-CAT4 and FR-CAT2 schemes that are described in [3, 9, 10].

We used the two dimension version MSP-CACRR(d1, d2) which uses the time intervals and the type of the day as the basic dimensions. PCAC-RR is a predictive technique that we introduced in [3] which utilizes the time interval and calculates the probabilities of finding the frequent full path. PR-CAT4 is a predictive scheme and it takes into account the movement to the next cell from the previous cell through the current cell. In FR-CAT2 scheme, there is no prediction and a fixed portion of the total bandwidth is reserved for the handoff calls. The results are shown in Fig. 4(a,b, and c). As we can see from these results, MSP-CACRR(d1, d2) has the overall best performance. From these figures, we can say that the predictive techniques have better performance in general compared to the non-predictive techniques. We can see also that there is a significant enhancement in the performance of
the MSP-CACRR($d_1, d_2$) technique compared to the PCAC-RR. This is because of two main reasons. The first reason is that in the MSP-CACRR($d_1, d_2$) technique we use an additional dimension to differentiate between the behavior of the user during the weekend and weekday, while only the time intervals was used in the PCAC-RR. This has significantly affected the performance of resource management. The second reason is that in the MSP-CACRR($d_1, d_2$) technique, we have used more efficient sequence mining technique (which is the MobilePrefixSpan technique) to extract the frequent sequences. This enables us to discover new frequent sequences that were not discovered in the PCAC-RR as explained before.

To study the prediction quality of the MSP-CACRR($d_1, d_2$) technique, we compared its performance with a Benchmark scheme has the same CAC and RR technique with important assumption that the prediction is perfect. This means that for the Benchmark scheme, we know exactly what will be next path for each user and what will be the exact handoff time. The results of the Benchmark scheme are also shown in Fig. 4. As we can see from these figures, the average performance of MSP-CACRR($d_1, d_2$) is about 85.27% of the average performance of the Benchmark. The average performance is the average of the three measures (CBP, CDP, and BWU). This ratio was about 70.65% in the PCAC-RR technique. This significant enhancement reflects that the MSP-CACRR($d_1, d_2$) has a high prediction quality. The main reason for that is using the MobilePrefixSpan technique and the new additional dimension of the type of the day.

VI. CONCLUSION

We have proposed a new technique based on multidimensional sequence mining called MSP-CACRR to solve the problem of CAC and RR for next generation cellular networks. This technique uses an efficient sequential pattern mining technique called MobilePrefixSpan. We used the time intervals of the day and the type of the day (weekend or working day) as the basic dimensions in this technique. Simulation results in terms of CBP, CDP, and BWU show that there is a significant improvement in the overall performance of the MSP-CACRR compared to other predictive technique. Comparing MSP-CACRR with Benchmark technique (which has a perfect prediction), it has been shown that the quality of prediction in the MSP-CACRR technique using the MoilePrefixSpan enables us to reach 85.27% of the average performance of the Benchmark technique. Our future research plan includes application of this predictive technique in the wireless heterogeneous networks where different type of mobile networks will be integrated to support the required services.

REFERENCES


[6] Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, Umeshwar Dayal, “Multi-dimensional sequential pattern mining”, Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, pp. 81-88, November 2001.


