ELSEVIER

Computer Networks 38 (2002) 577–589

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Mobility-based predictive call admission control and bandwidth reservation in wireless cellular networks ☆

Fei Yu, Victor Leung *,[1]

Department of Electrical and Computer Engineering, University of British Columbia, 2356 Main Mall, Vancouver, BC, Canada V6T 1Z4

Received 20 September 2001; accepted 15 October 2001

Responsible Editor: I.F. Akyildiz

Abstract

This paper presents call admission control and bandwidth reservation schemes in wireless cellular networks that have been developed based on assumptions more realistic than existing proposals. In order to guarantee the handoff dropping probability, we propose to statistically predict user mobility based on the mobility history of users. Our mobility prediction scheme is motivated by computational learning theory, which has shown that prediction is synonymous with data compression. We derive our mobility prediction scheme from data compression techniques that are both theoretically optimal and good in practice. In order to utilize resource more efficiently, we predict not only the cell to which the mobile will handoff but also when the handoff will occur. Based on the mobility prediction, bandwidth is reserved to guarantee some target handoff dropping probability. We also adaptively control the admission threshold to achieve a better balance between guaranteeing handoff dropping probability and maximizing resource utilization. Simulation results show that the proposed schemes meet our design goals and outperform the static-reservation and cell-reservation schemes. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Call admission control; Bandwidth reservation; Mobility prediction

1. Introduction

Future mobile communication systems are required to support broadband multimedia services with diverse quality of service (QoS) requirements. To utilize the radio spectrum efficiently, the cellular architecture is used in wireless networks. Since mobile users may change cells a number of times during the lifetime of their connections, availability of wireless network resources at the connection setup time does not necessarily guarantee that wireless network resources are available throughout the lifetime of a connection. Thus users may experience performance degradations due to mobile handoffs. This problem will be magnified in future micro/pico-cellular networks [10], where handoff events may occur at a much higher

☆ This work was supported by a grant from Motorola Canada Ltd., and by the Canadian Natural Sciences and Engineering Research Council under grant CRDPJ 223095.

* Corresponding author. Tel.: +1-604-822-6932; fax: +1-604-822-5949.

E-mail addresses: feiy@ece.ubc.ca (F. Yu), vleung@ee.ubc.ca (V. Leung).

[1] This paper is based on a paper presented at IEEE Infocom 2001, Anchorage, Alaska, April 2001.

1389-1286/01/$ - see front matter
PII: S1389-1286(01)00269-9

rate compared to today's macro-cellular systems. Call admission control (CAC) and bandwidth reservation mechanisms are required to address this problem. Since forced call terminations due to handoff blocking are generally more objectionable than new call blocking, in this paper we consider P_{hd}, the probability of handoff dropping, as the key connection-level QoS metric provisioned by CAC in wireless cellular networks. As it is impractical to completely eliminate handoff call dropping, the best one could do is to keep P_{hd} below a target level. Moreover, maximizing resource utilization while keeping P_{nb}, the probability of new call blocking, below a target value, is another critical factor for evaluating CAC algorithms.

Based on the above considerations, several schemes have recently been proposed for CAC in wireless cellular networks. The guard channel policy [15] and fractional guard channel policy [16] determine the number of guard channels reserved for handoffs by considering just the status of the local cell. Users are assumed uniformly located in any cell of the mobile network under these policies. The distributed CAC scheme [13] considers not only the status of the local cell but also that of adjacent cells. The total required bandwidth for both handoff and existing connections is calculated under the assumptions of exponentially distributed channel holding time and perfect knowledge of the rate of handoff. These assumptions are unrealistic in real networks. The shadow cluster scheme [11] estimates future resource requirements in a collection of cells in which a mobile is likely to visit in the future. Admission control is performed based on this estimate. However, this proposal lacks a mechanism to determine the shadow cluster in real networks, as it assumes either precise knowledge of user mobility or totally random user movements.

There have also been some research efforts to predict user mobility. In Ref. [12], the next cell to which a mobile will move is predicted in an indoor environment. But this scheme does not estimate channel holding time and therefore cannot be directly applied for efficient bandwidth reservation. In Ref. [5], handoff histories of mobile users are observed over time, and the mobile's movement can be predicted by utilizing this observation. However, in this scheme, prediction of each specific mobile's movement is based on the aggregate history of all users, and may not be accurate for each individual user.

Under more realistic assumptions, we propose CAC and bandwidth reservation schemes based on the probabilistic prediction of each individual user's movements. Our mobility prediction approach is derived from data compression techniques that are both theoretically optimal and good in practice. Our motivation is recent research work in computational learning theory [3,18], which has shown that prediction is synonymous with generalization and data compression. Similar prediction approaches were applied previously to the problems of prefetching in large-scale database system [21] and location management of mobile users in cellular networks [2]. Although the possibility of applying these approaches to QoS provisioning was mentioned in Ref. [2], no further development was reported. In this paper, we extend that idea to the context of CAC and bandwidth reservation. Our mobility prediction approach is novel in that we predict not only to which cell a mobile terminal will handoff but also when the handoff will occur.

The rest of this paper is organized as follows. System models employed in this paper that are more realistic than those considered previously in similar work are presented in Section 2. In Section 3, we describe and analyze our novel mobility prediction scheme. Based on the mobility prediction, efficient CAC and bandwidth reservation schemes are proposed in Section 4. Simulation results are presented and discussed in Section 5 demonstrating the effectiveness of our approach. Finally, we conclude this study in Section 6.

2. Model description

We consider a mobile communication network with a cellular wireless infrastructure. A handoff could fail due to insufficient bandwidth in the new cell, causing the handoff call to be dropped. In this paper, we do not consider (1) soft handoff in code division multiple access systems [20], in which a mobile can communicate with two base stations simultaneously; (2) delay-insensitive applications,

which can tolerate long handoff time delay when there is momentarily insufficient bandwidth. We describe the network topology, channel holding time distribution and user mobility pattern considered in our study in the following subsections.

2.1. Network topology

In solving the CAC and bandwidth reservation problems, most researchers model cellular networks by structured graphs. Circular, hexagonal or square cell configurations are often used in two-dimensional models, and a linear model is commonly used in the one-dimensional case [11,13]. Although these network topologies simplify the analyses, they do not accurately represent a real cellular network, where the number of neighboring cells varies from cell to cell, and the shape and size of each cell may vary depending on the receiver sensitivity, antenna radiation pattern of the base station, and the propagation environment.

Our network topology model is not restricted to a structured cell configuration such as hexagonal or linear. We use a generalized graph model to represent the actual cellular network. The network is modeled as a connected graph $G = (V, E)$, where the vertex-set V represents the set of base stations, each serving a single cell, and the edge-set E represents the adjacency between pairs of cells. Fig. 1 shows an example of a generalized graph model of a network representation with vertex-set $V = \{a, b, c, \ldots, l\}$ and edge-set $E = \{(a, b), (a, c), \ldots, (k, l)\}$.

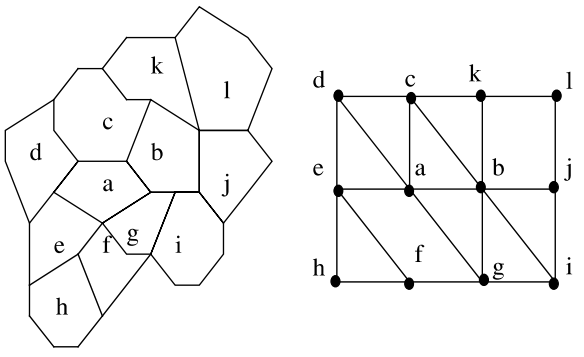

Fig. 1. Modeling an actual cellular network.

2.2. Channel holding time

The channel holding time is defined as the time during which a new or handoff call occupies a radio channel in a given cell, and it is dependent on the mobility of the user. While this is similar to the call holding time in the fixed telephone network, it is often a fraction of the total call duration in a wireless cellular network and needs not have the same statistical properties [7,8]. Most research work on CAC and bandwidth reservation assumes the channel holding times in all cells are independent and identically distributed (i.i.d.) according to an exponential distribution [6,11,16]. Like the structured models for network topology, i.i.d. exponential distribution simplifies the analyses, but does not give an accurate representation of the real characteristics of cellular networks.

We assume that the channel holding time follows a general distribution, which allows the i.i.d. exponential channel holding time assumption to be relaxed.

2.3. User mobility pattern

The *symmetric random walk model* has been quite popular among researchers in characterizing individual movement behavior [13,16]. In such a model, a mobile user will move to any one of the neighboring cells with equal probability after leaving a cell. This model does not take into account the trajectory and channel holding time of a mobile.

In cellular mobile networks, the mobility of a user during a call can be represented by a sequence of events, $N, H_1, H_2, H_3, \ldots, H_n, \ldots, E$, where N represents the event that a new call is admitted, H_n represents the event of a mobile's nth handoff and E represents the call termination event. Note that in some cases, there are no handoff events during the lifetime of a call and thus no H_n in the sequence of events. In this sequence, $N = (m, i, t)$, where m represents the mobile requesting the call, i represents the original cell and t represents the time when the call arrives; $H_n = (T_k, i)$, where T_k is the relative time elapsed since the beginning of the call and i is the cell to which the mobile will handoff; and $E = (T_k)$. We quantize the relative time into

slots of equal duration T, a design parameter. So, T_k is the kth time slot since the beginning of the call.

In general, a mobile user usually travels with a specific destination in mind. So, the mobile's location and channel holding time in the future are likely to be correlated with its movement history. Therefore, in our model, the sequence of events $N, H_1, H_2, H_3, \ldots, H_n, \ldots, E$ is assumed to be generated by an *mth order Markov source*, in which the states correspond to the contexts of the previous m events. The probabilities of possible next events can depend on a list of m previous events.

3. Mobility prediction

We derive probabilistic prediction of user mobility based on the accumulated behavior history of each specific mobile. The rationale behind the prediction scheme is the observation that a user's mobility pattern is a reflection of the routines of his/her life and most mobile users have favorite routes and habitual movement patterns. This repetitive nature of mobility patterns suggests the *stationarity* of a sequence of events generated by an mth order Markov source. Then, we can learn those patterns from the mobility history of the respective user and predict the user's next move when those patterns reappear.

Similar prediction approaches is used in Ref. [2] to solve the location management problem in cellular networks. The proposed method records only the locations of mobile users to predict their future locations [2]. This method cannot be used directly to derive efficient CAC and bandwidth reservation schemes. Although the possibility of using this method for QoS provisioning in cellular networks is mentioned in Ref. [2], as far as we know our proposal is the first to realize this possibility. The novelty of our proposal compared to the previous proposal [2] is that we record both the locations and the handoff times of the mobile users. Therefore, we can derive a novel prediction method that predicts not only where a mobile user will handoff but also when the handoff will likely occur. Based on this novel prediction method, we further propose CAC and bandwidth reservation schemes that are more efficient than existing methods.

The prediction approach is motivated from optimal data compression methods. In data compression, a data set (e.g., a text file or an image) is decomposed into a sequence of events, and encoded using as few bits as possible. Thus, short codewords should be assigned to more probable events and longer codewords should be assigned to less probable events. So, in order to compress data well, one has to be able to predict future data well, and hence a good data compressor should also be a good predictor. If a data compressor expects a certain character to be next with a very high probability, it will assign that character a relatively short code. If the overall code length is small, then the predictions of the data compressor must have been good.

3.1. Optimal data compression

In this paper, we develop our mobility prediction algorithm based on the Ziv–Lempel algorithms for data compression, which are both theoretically optimal and good in practice. The original word-based Ziv–Lempel encoder [23] breaks the input string into block-to-variable codes. The algorithm parses each block of size n in a greedy manner into distinct substrings x_1, $x_2, \ldots, x_n$ with the following property. For each $j \geqslant 1$, substring x_j without its last character is equal to some previous substring x_i, where $0 \leqslant i < j$. Substring x_j is encoded by the value i, using $\lceil \lg(j-1) \rceil$ bits[2], followed by the ASCII encoding of the last character of x_j, using $\lceil \lg \alpha \rceil$ bits, where α is the size of the input sequence's alphabet. Because of this *prefix property*, the substring parsed so far can be efficiently maintained in a trie [9].

The equivalent character-based Ziv–Lempel algorithm builds in an on-line fashion a probabilistic model (or a trie) that feeds probability information to an arithmetic coder [22], which encodes a sequence of probability of p using $\lg(1/p) = -\lg p$ bits. We show by an example how these algorithms work.

[2] The base of the logarithm is 2 in this paper.

Example 1. Let the alphabet be $\{a, b, c\}$. We consider an input string "*aababcbaccababa*..." that the Ziv–Lempel encoder parses as "$(a)(ab)(abc)(b)(ac)(c)(aba)(ba)\ldots$." Each substring in the parse is encoded as a pointer followed by an ASCII character. In particular, the match "*ab*" of the seventh substring "*aba*" is encoded using $\lceil \lg(6) \rceil$ bits with a value 2, since "*ab*" matches the second substring, and the last character "*a*" is encoded using $\lceil \lg 3 \rceil$ bits, since the alphabet size is 3.

In the character-based version of Ziv–Lempel encoder, a trie is built when the previous substring ends. The trie at the start of the ninth substring is pictured in Fig. 2. There are five previous substrings beginning with an "*a*", two beginning with an a "*b*" and one beginning with a "*c*". The character "*a*" is therefore assigned a probability of $\frac{5}{8}$ at the root, "*b*" is assigned a probability of $\frac{2}{8}$ at the root, and "*c*" is assigned of probability of $\frac{1}{8}$ at the root. Similarly, of the five substrings that begin with an "*a*", three begin with an "*ab*" and one begins with an "*ac*", giving the probability of $\frac{3}{5}$ for "*b*" and $\frac{1}{5}$ for "*c*" at node $\{a, [5/8]\}$, and so on. Any sequence that leads from the root of the trie to a leaf traverses a sequence of probabilities of $p_1, p_2, p_3, \ldots$, whose product $\prod p_i$ equals $\frac{1}{8}$. The arithmetic coder encodes the sequence with $-1\mathrm{g}\prod_i p_i = 1\mathrm{g}\ 8 = 3$ bits. Note that the square nodes in Fig. 2 denote the last nodes ending the sequence.

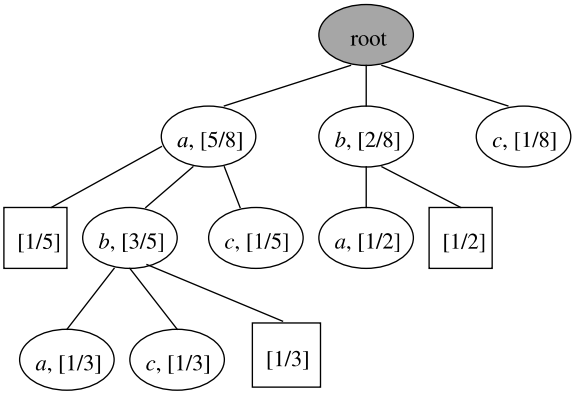

Fig. 2. The trie constructed in Example 1.

3.2. Mobility prediction

Our mobility prediction scheme is based on the character-based version of the Ziv–Lempel algorithm. The sequence of events $N, H_1, H_2, H_3, \ldots, H_n, \ldots, E$ during the lifetime of a call corresponds to a substring in the Ziv–Lempel algorithm. The mobility database of every mobile at a specific time holds a *mobility trie*, which is a probability model corresponding to that of the Ziv–Lempel algorithm. Each node except for the root in the mobility trie preserves the relevant statistics that can be used to predict the probability of following events. As in data compression, the mobility trie of the mobile is built in an on-line fashion. When a mobile requests a new call, the predictor sets the current node to the root of the trie according to the identity of the mobile, the cell it is in, and the current time, and calculates the probabilities of all possible events of this mobile. Upon recording an actual event of the mobile, the predictor walks down the trie and is ready for the next prediction. When an event is not in the mobility trie, a prediction fault is generated and the trie is updated accordingly. A pseudocode description of the mobility prediction scheme is given in Fig. 3.

Fig. 4 shows an example mobility trie of mobile *m* at cell *a* in the time interval 9:00–9:01 a.m. When the mobile requests a new call in cell *a* in the time interval 9:00–9:01 a.m., we can use the statistics preserved in the nodes of its mobility trie to predict

```
initialize mobility trie :=null
   initialize number_of_event :=0
   loop
      wait for next event e
      if (e is a new call)
          look for a trie with a root of e in the database
            if (can not find such a trie)
               create a trie :=single node (the root)
            endif
      else
         if (e exists in the trie)
           number_of_event :=number_of_event + 1
         else
            create a leaf  e
      calculate the probabilities of possible events
             based on the number_of_event of leaves
  forever
```

Fig. 3. Pseudocode of mobility prediction.

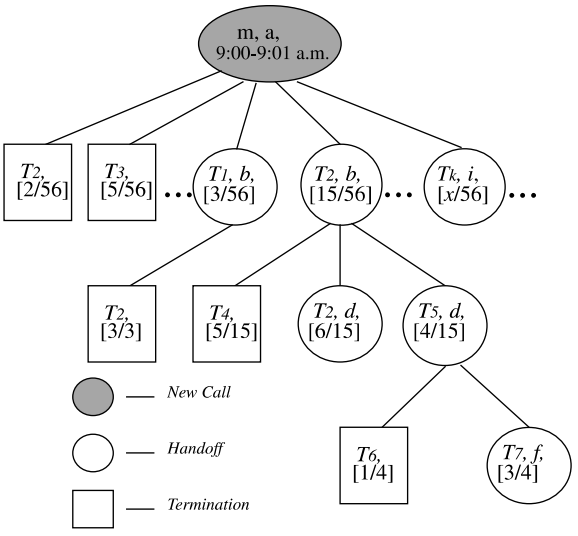

Fig. 4. A mobility trie used for mobility prediction.

the probabilities of the next possible events of this mobile: it will terminate the call without handoffs in the second time slot with probability of 2/56, handoff to cell b in the second time slot with probability of 15/56, etc.

3.3. Analysis of the mobility prediction scheme

We first analyze the optimality of word-based Ziv–Lempel algorithm and show that the character-based Ziv–Lempel algorithm is as least as good as the word-based approach. We then establish that our mobility prediction scheme inherits the optimality of these data compression algorithms.

Given a sequence x^n of length n over an alphabet A of α letters and an *information lossless* (IL) compressor C accepting inputs over A, let $|C(x^n)|$ denote the length, in bits, of the output that C produces on x^n. The compression ratio $\rho_C(x^n)$ attained by C for x^n is [23]:

$$\rho_C(x^n) = \frac{|C(x^n)|}{n \lg(\alpha)}. \tag{1}$$

Define $\rho_\sigma(x^n)$ as the best compression ratio attainable for x^n by any IL compressor of σ states. Sequence x^n is parsed into different phrases: $x^n = x_1, x_2, \ldots, x_t$. The maximum possible number of distinct phrases is $t(x^n)$. Define

$$q(x^n) = \frac{t(x^n)\ \lg(t(x^n))}{n \lg(\alpha)}. \tag{2}$$

A result in Ref. [23] shows that

$$\rho_\sigma(x^n) \geqslant q(x^n) - \delta(\sigma, n) \quad \text{with } \lim_{n\to\infty} \delta(\sigma, n) = 0. \tag{3}$$

So, $q(x^n)$ is a lower bound on the compression ratio attainable for x^n by any codebook.

The Ziv–Lempel incremental parsing algorithm achieves for any sequence x^n given to it a compression ratio that is (asymptotically) equal to $q(x^n)$, and thus the algorithm is universal and asymptotically optimal [17].

In Ref. [1], it has been shown that the code length obtained in the character-based version of the Ziv–Lempel algorithm is as least as good as that obtained using the word-based approach. Hence, the optimality result in Ref. [23] holds without change for the character-based approach.

We define the *event fault rate* to be the total number of event faults incurred by our mobility prediction algorithm divided by the total number of events. Also, we define the *expected event fault rate* to be the best possible event fault rate achievable by any prediction algorithm which makes its prediction based only on the past history.

A result in Ref. [21] shows that, if the source is a stationary mth order Markov source, the expected event fault rate of our prediction algorithm is within an additive factor of $O(1/\sqrt{n})$ from the expected event fault rate of the source, where n is the length of the event sequence.

From these, we see that our mobility prediction algorithm inherits the asymptotic optimality of the Ziv–Lempel algorithm. By modeling the sequence of events during the lifetime of a call as that generated by a stationary mth order Markov source and predicting next events using the mobility prediction scheme derived from the Ziv–Lempel algorithm, we can predict not only to which cell a mobile will handoff but also when the handoff will occur.

3.4. Implementation considerations

The mobility prediction scheme proposed above maintains the statistics in a trie. An important issue is how this model can be implemented. In fact, a trie is a multiway tree with a path from the root to a unique node for each string represented in the tree. There are many ways to implement the nodes of a trie. The fastest approach for processing is to create an array of pointers for each node in the trie with a pointer for each character of the input alphabet (Fig. 5a). This method can waste considerable memory space, particularly if some characters of the alphabet are rarely used. An alternative is to use a linked list at each node, with one item for each possible branch (Fig. 5b). This uses memory economically, but can be more processing intensive. Some improvement may be achieved by moving an item to the front of the list each time it is used. A trie can also be implemented as a single hash table with an entry for each node. The memory consumed by a trie can be reduced by truncating it prematurely at a shallow depth, and using some other data structure for subsequent characters. For further details, the reader can consult books on algorithms and data structures.

In practice, in order to reduce the memory and computation complexity, it is desirable to limit the size of the data structure for prediction. Several techniques are known for limiting data structure size [19]. An explicit upper bound M is placed on the size of the data structure. The data structure is either frozen when its size reaches M, flushed and rebuilt when its size reaches M, or frozen when its size reaches $M/2$ and a new one is built while the old one is used for prediction. There are also more sophisticated techniques that use least-recently-used (LRU) strategy [4] on the data structure to maintain its size. In our simulations in Section 5, we set upper bound 200 bytes to the trie size and use LRU strategy to maintain its size.

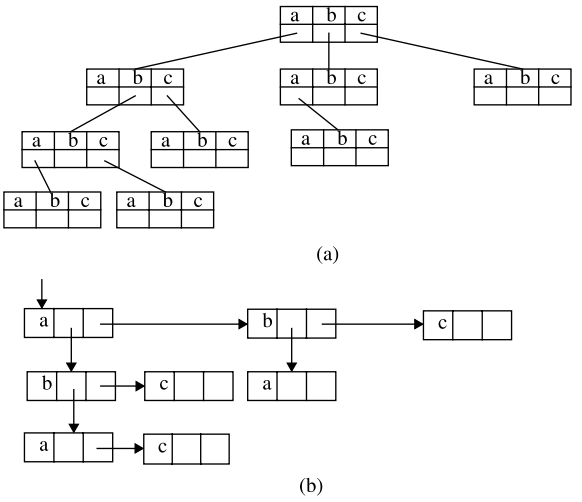

Fig. 5. Implementation of trie nodes for Fig. 2: (a) array and (b) linked list.

4. Call admission control and bandwidth reservation

4.1. Calculation of $P_{i,j}(T_k)$

Our approach is based on the predicted mobility of each user. We calculate $P_{i,j}(T_k)$, the probability that a mobile originally in cell i will visit cell j during time slot T_k. From the mobility trie, we can see that a mobile taking different paths can visit certain cell in the same slot. Using the *total probability theorem* [14], we must add all of these probabilities to get $P_{i,j}(T_k)$. We show by an example how to get this probability.

Example 2. A mobile m requests a new call at cell a in the time interval 9:00–9:01 a.m. From the mobility trie in Fig. 4, we can see that m can take several different paths to visit cell b. We describe these paths by sequences of events:

Path 1: $N(m, a, 9{:}00\text{–}9{:}01\ \text{a.m.}), H(T_1, b), E(T_2)$.

By path 1, m will visit cell b in T_1 and T_2 with probability: $\frac{3}{56} \times \frac{3}{3} = \frac{3}{56}$.

Path 2: $N(m, a, 9{:}00\text{–}9{:}01\ \text{a.m.}), H(T_2, b), E(T_4)$.

By path 2, m will visit cell b in T_2, T_3 and T_4 with probability: $\frac{15}{56} \times \frac{5}{15} = \frac{5}{56}$.

Path 3: $N(m, a, 9{:}00\text{–}9{:}01\ \text{a.m.}), H(T_2, b), H(T_2, d) \ldots$

By path 3, m will visit cell b in T_2 with probability: $\frac{15}{56} \times \frac{6}{15} = \frac{6}{56}$.

Path 4: $N(m, a, 9{:}00\text{–}9{:}01\ \text{a.m.}), H(T_2, b), H(T_5, d) \ldots$

By path 4, m will visit cell b in T_2, T_3, T_4 and T_5 with probability: $\frac{15}{56} \times \frac{4}{15} = \frac{4}{56}$.

So,

$$P_{a,b}(T_1) = \frac{3}{56},$$
$$P_{a,b}(T_2) = \frac{3}{56} + \frac{5}{56} + \frac{6}{56} + \frac{4}{56} = \frac{18}{56},$$
$$P_{a,b}(T_3) = \frac{5}{56} + \frac{4}{56} = \frac{9}{56},$$
$$P_{a,b}(T_4) = \frac{5}{56} + \frac{4}{56} = \frac{9}{56}$$

and

$$P_{a,b}(T_5) = \frac{4}{56}.$$

4.2. The most likely cell-time

When a mobile is active in cell i, we can get the most likely cell-time (MLCT) of that mobile, a cluster of time units at a cluster of cells when and where a mobile will most likely visit in the future. We select cell j and time slot T_k with $P_{i,j}(T_k)$ greater than zero to form the MLCT of this mobile.

4.3. Bandwidth reservation

Using $P_{i,j}(T_k)$, the probabilities of handing off from cell i into cell j during time slot T_k of mobile m, we can obtain the required bandwidth $B_{\text{reserved}}(j, T_k, m)$ to be reserved in cell j for the expected handoff of m from cell i:

$$B_{\text{reserved}}(j, T_k, m) = P_{i,j}(T_k)B(m), \tag{4}$$

where $B(m)$ is the bandwidth required by m. Moreover, the reserved bandwidth $B_{\text{reserved}}(j, T_k)$, which is the aggregate bandwidth to be reserved in cell j during T_k, is calculated as

$$B_{\text{reserved}}(j, T_k) = \sum_{m \in M} B_{\text{reserved}}(j, T_k, m), \tag{5}$$

where M is the set of mobiles which will handoff to cell j from cell i during T_k. Finally, the free bandwidth left after the reservation is

$$B_{\text{free}}(j, T_k) = B - B_{\text{reserved}}(j, T_k), \tag{6}$$

where B is the total bandwidth in cell j.

4.4. Call admission control and bandwidth reservation for new calls

When a new call arriving at mobile m with a bandwidth requirement $B(m)$ requires admission to cell i, the CAC algorithm first checks if the current free bandwidth of cell i can support the call. The call is rejected if the cell does not have enough free bandwidth. Otherwise, the CAC algorithm will check the availability of free bandwidth in the MLCT of this mobile. The checking result can be written as

$$\text{Check}(j, T_k, B(m)) = \begin{cases} 1, & B_{\text{free}}(j, T_k) \geqslant B(m), \\ \dfrac{B_{\text{free}}(j, T_k)}{B(m)}, & \text{otherwise.} \end{cases} \tag{7}$$

Based on these values, the new call will be admitted if the following holds:

$$\sum_{j,k \in \text{MLCT}} P_{i,j}(T_k)\text{Check}(j, T_k, B(m)) \geqslant \alpha \sum_{j,k \in \text{MLCT}} P_{i,j}(T_k), \tag{8}$$

where α is the *admission threshold* and should be controlled adaptively. We will describe how to control this threshold in the next subsection.

When a new call is admitted, bandwidth is reserved in the mobile's MLCT, and the free bandwidth in the MLCT is updated accordingly.

4.5. Adaptive control of admission threshold

The mobility prediction functions may not work well for some mobile users, especially those who do not have favorite routes. Moreover, if the admission threshold α is too small, the handoff dropping probability may exceed the target value; if α is too large, resource utilization will be decreased. So, admission threshold α should be controlled adaptively.

We calculate $P_{\text{hd}}(m)$, the handoff dropping probability of mobile m, by dividing the number of handoff drops to the total number of its calls recorded in the mobility trie. Let $P_{\text{hd,target}}(m)$ denote the target value of handoff dropping probability of mobile m. If $P_{\text{hd}}(m) < P_{\text{hd,target}}(m)$, admission threshold α is decreased by ε, a design parameter;

otherwise α is increased by ε. The calculation of $P_{hd}(m)$ and update of the admission threshold are done upon call completion.

By adaptive control of α, we can achieve a better balance of guaranteeing P_{hd} and maximizing resource utilization.

4.6. Call admission control and bandwidth reservation for handoff calls

When a mobile m, with bandwidth requirement $B(m)$, requires a handoff to cell i, the CAC algorithm will admit it if the current free bandwidth of cell i can support the call. Then, the CAC algorithm will calculate $P_{i,j}(T_k)$ and get the MLCT of m based on the mobility trie. Bandwidth is reserved for m in its MLCT accordingly.

5. Simulation results and discussion

In this section, we present and discuss the simulation results of the proposed schemes as well as the comparisons with two other CAC schemes.

We consider a coverage area that consists of 40 base stations, each having six neighbors on average. The average distance between two base stations is 1 mile. Since most mobile users have favorite routes, we assume that each mobile user has five possible different paths in the network. The user will take these five paths with probability of 0.5, 0.2, 0.1, 0.1, 0.1, respectively. Among the cells within a path, mobile users can have a new call request with equal probabilities. During a call, the mobile will stay at the original cell or move along the path. If a call does not terminate when the mobile reaches the end of the path, it will stay at the end cell of that path. The path is generated as follows: (1) Select two nodes in the graph randomly as original and destination nodes. (2) Whenever the mobile user leaves the current cell, it moves to a neighboring cell which is closest to the destination. Note that two paths with at least one edge not in common are different paths and different mobile users can have the same paths.

Also, we apply the following assumptions in our model:

1. Each cell has a fixed link capacity of 40 bandwidth units (BUs).
2. Time is quantized into units of $T = 30$ s.
3. A call is either for voice (requiring 1 BU) or video (requiring 4 BUs).
4. Call durations are the same for all calls and exponentially distributed with mean value of 120 s.
5. Call requests are generated according to a Poisson process with rate (calls/cell/s).
6. Two cases of mobility are considered: low user mobility, in which case the speeds of mobiles are uniformly distributed between 0 and 40 miles/h; and high user mobility, in which case the speeds of mobiles are uniformly distributed between 40 and 70 miles/h.
7. The target handoff dropping probabilities are the same for all mobiles: $P_{hd} = 0.01$.
8. Admission threshold α is initialized to 1 in each simulation and adaptive factor $\varepsilon = 0.02$.

The offered load is calculated as follows:

$$\text{Offered Load} = 120\ \lambda\ ((1 - P_{\text{voice}})4 + P_{\text{voice}}),$$

where P_{voice} is the percentage of voice calls in the offered traffic.

Simulations start without any pre-memorized information of mobiles. Long-term handoff dropping probability, new call blocking probability and utilization are obtained for a 100 h simulation time duration. During each simulation, a mobility tries is constructed for each mobile and its mobility is predicted. Based on the prediction, a MLCT is constructed. Then CAC algorithm will check the availability of bandwidth and decide to admit or reject the new call and handoff call requests using the algorithms described in Section 4. If a call is admitted, bandwidth is reserved in the mobile's MLCT accordingly.

Figs. 6 and 7 show P_{nb} and P_{hd} as functions of the offered load for two values of P_{voice}:0.8 and 1 in the low mobility and high mobility cases. The probabilities of handoff dropping are kept below the target values 0.01 irrespective of the offered load, P_{voice} and user mobility. This shows that the proposed CAC and bandwidth reservation schemes achieve one of our goals: keeping P_{hd}

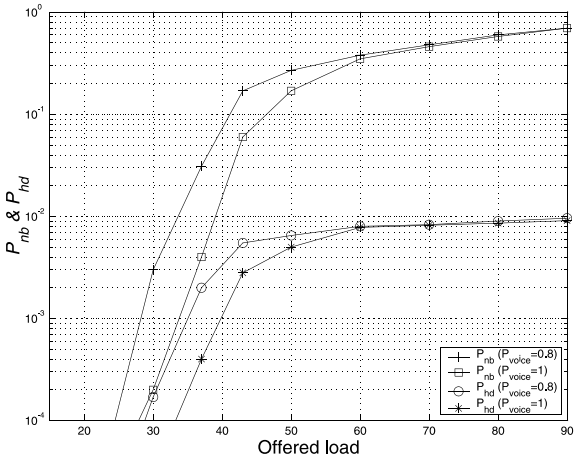

Fig. 6. P_{nb} and P_{hd} vs. offered load (low user mobility).

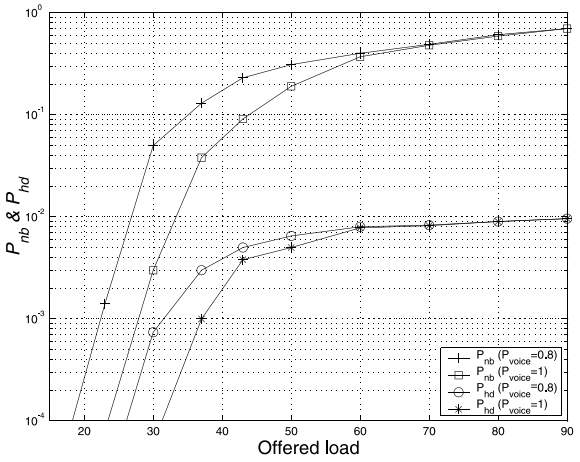

Fig. 7. P_{nb} and P_{hd} vs. offered load (high user mobility).

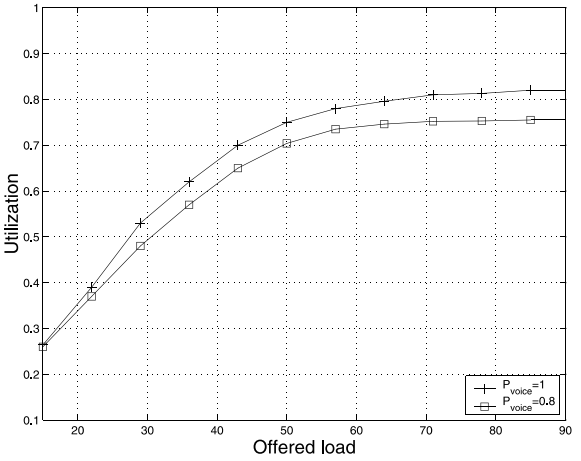

Fig. 8. Utilization vs. offered load.

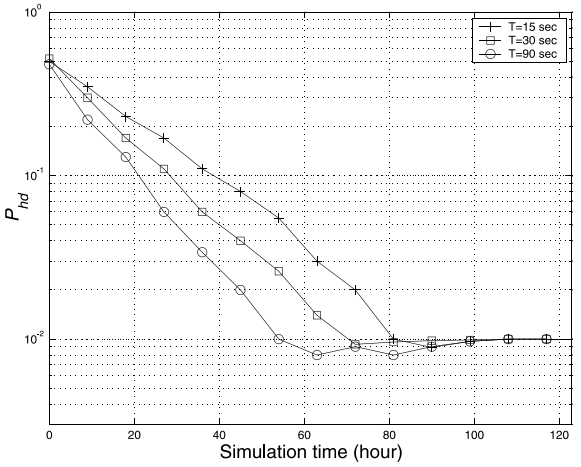

Fig. 9. P_{hd} vs. simulation time with different values of time slot duration.

below a target level. We also observe that the P_{nb} and P_{hd} increase as P_{voice} decreases under the same offered load. This is because the video connections need more bandwidth. Fig. 8 shows the average utilization as a function of the offered load in the low user mobility case.

Since the time slot is used in our mobility prediction scheme, the selection of time slot duration T will have influence on both the convergence speed and the network utilization. We study this issue in the following. In simulations, we choose three values of T, 15, 30 and 90 s. Fig. 9 shows the handoff dropping probabilities as functions of simulation time with different time slot durations. In the beginning of the simulation, the system knows little about the mobility of mobile users and the handoff dropping probabilities cannot be kept below the target value 0.01. As the simulation goes on, the prediction tends to converge and the target handoff dropping probability can be guaranteed. From Fig. 9 we can see that the convergence is faster when T is 90 s than when T is 30 s, which is in turn faster than when T is 15 s. The reason is that the mobility sequence will be long when T is small, and when T is large, the sequence will be short. The longer the sequence, the slower the convergence speed. It seems that it is better to have

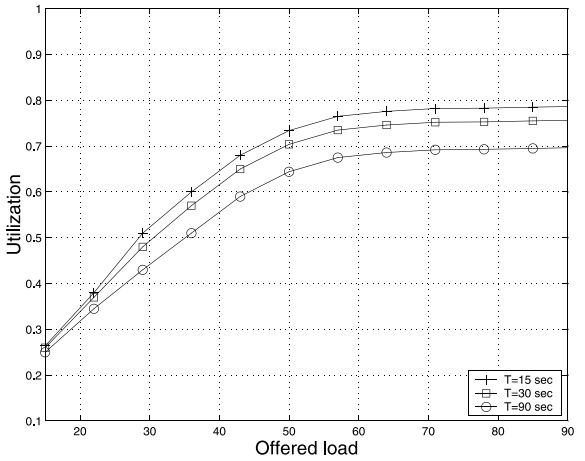

Fig. 10. Utilization vs. offered load with different values of time slot duration.

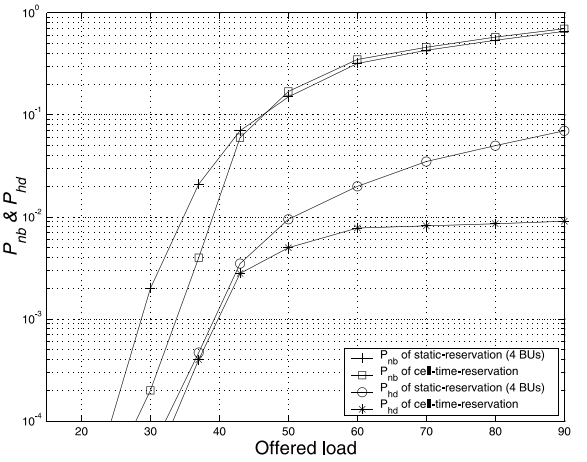

Fig. 11. Comparison with static reservation (4 BUs).

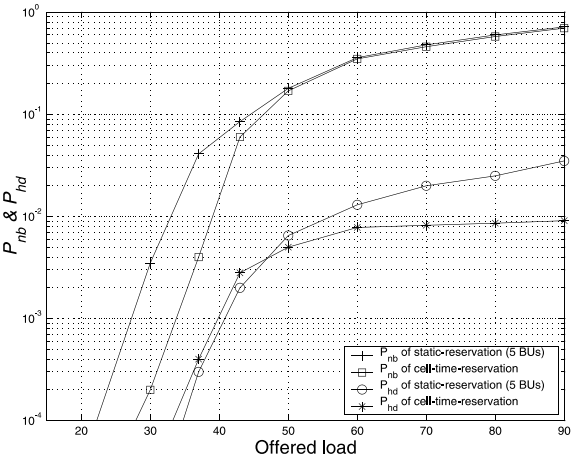

Fig. 12. Comparison with static reservation (5 BUs).

large value of T, since the convergence will be fast. However, large value of T means that the prediction is not accurate and will result in low utilization, which can be seen clearly in Fig. 10. The utilization is higher when T is 15 s than when T is 30 s, which is higher than when T is 90 s. Therefore, choosing a suitable value of T according to the real network condition is very important to get the best performance from the proposed scheme.

We also compare the proposed CAC and bandwidth reservation schemes with two other schemes: (1) static reservation [15]; and (2) cell reservation [12]. In the static-reservation scheme, a set of bandwidth is reserved permanently for handoff calls. In our simulation, we consider 4 BUs and 5 BUs reserved permanently for handoff calls in each cell. In the cell-reservation scheme, only the location of the mobile user but not the visiting time is predicted. Bandwidth is reserved in those cells that the mobile will visit during the entire lifetime of the call. For comparison, we call our scheme cell-time reservation. In these comparisons, we set $P_{\text{voice}} = 0.8$ in the low mobility case.

Figs. 11 and 12 show that the static-reservation scheme with 4 and 5 BUs reserved for handoff calls can keep P_{hd} below the target value of 0.01 when the network has a light load, but the reserved bandwidth is not enough when the offered load becomes heavier. Hence, this scheme cannot achieve the design goal. Although the static-reservation scheme has almost the same P_{hd} compared with our scheme when the network load is lighter, its P_{nb} is higher in this area, i.e., it admits less new calls than our scheme for any given P_{nb}. In the static-reservation scheme, P_{hd} may be kept below the target value by permanently reserving more bandwidth for handoff calls. However, this will result in higher P_{nb}, which means lower utilization if P_{nb} were to be reduced to an acceptable level. Fig. 13 compares our cell-time reservation scheme with the cell-reservation scheme. We can see that the cell-reservation scheme can keep P_{hd} below the target value 0.01 at the expense of higher P_{nb} compared with our scheme. This is because our

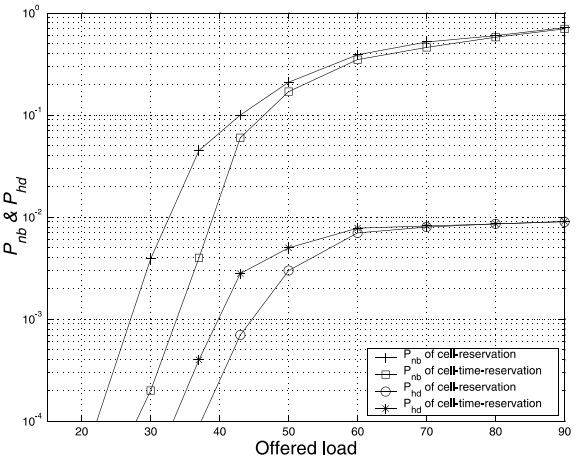

Fig. 13. Comparison with cell reservation.

scheme predicts not only to which cell the mobile will handoff but also when the handoff will occur. Based on the mobility prediction, we can reserve bandwidth more efficiently. From these results, we can see that our proposed cell-time reservation scheme achieves a better balance of guaranteeing P_{hd} and maximizing utilization.

6. Conclusions

In this paper, we have proposed CAC and bandwidth reservation schemes for wireless cellular mobile networks based on assumptions more realistic than existing proposals. The proposed schemes are applicable to arbitrary cell topologies and the channel holding time can follow a general distribution. The sequences of events of new call admission, handoffs and call termination are modeled by stationary mth order Markov sources. We derive novel probabilistic predictions of next events based on the mobility history of users, using an algorithm motivated by optimal data compression. Based on the mobility prediction of where and when the mobile will handoff to the next cell, CAC and bandwidth reservation schemes have been developed. The performance of the proposed schemes have been studied using computer simulations. Results show that our schemes can achieve a better balance of guaranteeing handoff dropping probability while maximizing resource utilization, and they outperform the static-reservation and cell-reservation schemes.

References

[1] T.C. Bell, J.C. Cleary, I.H. Witten, Text Compression, Prentice-Hall Advanced Reference Series, Prentice-Hall, Englewood Cliffs, NJ, 1990.
[2] A. Bhattacharya, S.K. Das, LeZi-update: an information theoretic approach to track mobile users in PCS networks, Proceedings of MOBICOM'99, Seattle, WA, August 1999.
[3] R. Board, L. Pitt, On the necessity of occam algorithms, Proceedings of the 22nd Annual ACM Symposium on Theory of Computation, New York, May 1990.
[4] S. Bunton, G. Borriello, Practical dictionary management for hardware data compression, Department of Computer Science, University of Washington, FR-35, 1991.
[5] S. Choi, K.G. Kin, Predictive and adaptive bandwidth reservation for hand-offs in QoS-sensitive cellular networks, Proceedings of ACM SIGCOMM'98, Vancouver, BC, September 1998.
[6] C. Chao, W. Chen, Connection admission control for mobile multiple-class personal communications networks, IEEE J. Select. Areas Commun. 15 (8) (1997) 1618–1626.
[7] Y. Fang, I. Chlamtac, A new mobility model and its application in the channel holding time characterization in PCS networks, Proceedings of IEEE INFOCOM'99, New York, March 1999.
[8] C. Jedrzycki, V.C.M. Leung, Probability distributions of channel holding time in cellular telephony systems, Proceedings of IEEE VTC'96, Atlanta, May 1996.
[9] G.G. Langdon, A note on Ziv–Lempel model for compressing individual sequences, IEEE Trans. Inf. Theory 29 (2) (1983) 284–287.
[10] W.C.Y. Lee, Smaller cells for greater performance, IEEE Commun. Mag. 29 (1991) 19–23.
[11] D.A. Levine, I.F. Akyildiz, M. Naghshineh, A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept, IEEE/ACM Trans. Network. 5 (1997) 1–12.
[12] S. Lu, V. Bharghavan, Adaptive resource management algorithms for indoor mobile computing environment, Proceedings of ACM SIGCOMM'96, Stanford, CA, September 1996.
[13] M. Naghshineh, M. Schwartz, Distributed call admission control in mobile/wireless networks, IEEE J. Select. Areas Commun. 14 (4) (1996) 711–717.
[14] A. Papoulis, Probability, Random Variables, and Stochastic Processes, third ed., McGraw-Hill, New York, 1991.
[15] E.C. Posner, R. Guerin, Traffic policies in cellular radio that minimize blocking of handoff calls, Proceedings of 11th Teletraffic Cong. (ITC-11), Kyoto, Japan, September 1985.
[16] R. Ramjee, R. Nagarajan, D. Towsley, On optimal call admission control in cellular networks, Proceedings of IEEE INFORCOM'96, San Francisco, CA, March 1996.

[17] D. Sheinwald, On the Ziv–Lempel proof and related topics, Proc. IEEE 82 (1994) 866–871.
[18] D.D. Sleator, R.E. Tarjan, Amortized efficiency of list update and paging rules, Commun. ACM 28 (2) (1985) 202–208.
[19] J.A. Storer, Data Compression Methods and Theory, Computer Science Press, Rockville, MD, 1988.
[20] A.J. Viterbi, CDMA: Principle of Spread Spectrum Communication, Addison-Wesley, Reading, MA, 1995.
[21] J.S. Vitter, P. Krishnan, Optimal prefetching via data compression, J ACM 43 (5) (1996) 771–793.
[22] I.H. Witten, R.M. Neal, J.G. Cleary, Arithmetic coding for data compression, Commun. ACM 30 (1987) 520–540.
[23] J. Ziv, A. Lempel, Compression of individual sequences via variable-rate coding, IEEE Trans. Inf. Theory 24 (1978) 530–536.

Fei Yu received the B.S. degree in Electrical Engineering from Dalian University of Technology, P.R. China, in 1995, and the M.S. degree in Computer Engineering from Beijing University of Posts and Telecommunications, P.R. China, in 1998. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, the University of British Columbia, Vancouver, Canada. His research interests are quality of service, mobility management and internetworking issues in wireless networks.

Victor C.M. Leung received the B.A.Sc. (Hons.) degree in Electrical Engineering from the University of British Columbia (UBC) in 1977, and was awarded the APEBC Gold Medal as the head of the graduating class in the Faculty of Applied Science. He attended graduate school at UBC on a Natural Sciences and Engineering Research Council Postgraduate Scholarship and obtained the Ph.D. degree in electrical engineering in 1981.

From 1981 to 1987, Dr. Leung was a senior member of Technical Staff at MPR Teltech Ltd., Canada, specializing in the planning, design and analysis of satellite communication systems. He also held a part-time position as visiting Assistant Professor at Simon Fraser University in 1986 and 1987. In 1988, he was a Lecturer in the Department of Electronics at the Chinese University of Hong Kong. He joined the Department of Electrical and Computer Engineering at UBC in 1989, where he is a Professor and holder of the TELUS Mobility Research Chair in Advanced Telecommunications Engineering. He is also a member of the UBC Institute for Computing, Information and Cognitive Systems. He is a Project Leader and member of the Board of Directors in the Canadian Institute for Telecommunications Research, a Network of Centres of Excellence funded by the Canadian Government. He is an Editor of the IEEE Transactions on Wireless Communications, and an Associate Editor of the IEEE Transactions on Vehicular Technology. His research interests are in the areas of architectural and protocol design and performance analysis for computer and telecommunication networks, with applications in satellite, mobile, personal communications and high speed networks. Dr. Leung is a senior member of IEEE and a voting member of ACM.