Packet Cellular System Optimizations for Non-Uniform Traffic

Paul Blair
pblair@ucsd.edu
Center for Wireless Communications
University of California, San Diego
La Jolla, CA 92093-0114, U.S.A.

George C. Polyzos
polyzos@aueb.gr
Department of Informatics
Athens University of Economics and Business
Athens 10434, Greece

Abstract—**An analysis of a Plane Cover Multiple Access cellular system in the presence of non-uniform traffic is presented. PCMA seeks to maximize the number of parallel transmissions among cells by defining virtual cells in which users transmit using different reuse factors. It is demonstrated how PCMA is novel in its ability to more efficiently service users in a system with non-uniform traffic load among cells and to adapt in real-time to changes in traffic distribution. After formally stating the optimal coverage problem, a greedy algorithm is presented and used to improve substantially over PCMA with simple, uniform resource allocation.**

I. Introduction

PLANE Cover Multiple Access (PCMA) has previously been studied as a means of increasing system capacity in a wireless, cellular system [1]. Because traditional TDMA wireless systems have provided low performance in terms of system throughput, such alternative schemes as PCMA need to be explored. Current systems such as GSM use static resource allocation schemes such as one-seventh frequency reuse. Consequently, they are inefficient in terms of system capacity since only a fraction, such as $1/7$, of resources are available in a given cell. Alternative schemes such as CDPA [2] have been proposed for packet environments to improve on attainable capacity. However, it has been shown that PCMA can provide even greater system capacity under a variety of propagation conditions and can also provide minimal delay relative to alternative systems [1]. In this paper it will be demonstrated how PCMA is also preferable in its ability to adapt to changing traffic distributions among system cells and to provide greater user capacity subject to a given Quality of Service (QoS) than competing schemes.

The most novel feature of PCMA is its use of Virtual Cells (VCs). A virtual cell is by definition an area of the system in which all users transmit using the same reuse factor. Virtual cells can simply be concentric sub-cells of system cells, with the base station of that system cell servicing all mobiles in the VC, or they could cross system cell boundaries [3]. In the latter case, several base stations will induce the virtual cell by cooperatively servicing users from the virtual cell.

In the current study we investigate ways in which VCs permit dynamic system adaptations that respond favorably to non-uniform traffic load and changes in the load stemming from mobility. In particular, we look at how such adaptations can help to better maintain QoS guarantees.

Such a study is of paramount importance in assessing cellular systems, since all real systems present the challenge of non-uniform traffic to the designer. The reason stems from user mobility in a real-time, unpredictable fashion. However, even if all users were stationary, traffic intensity would still vary widely from cell to cell due to the time-varying nature of user demands for bandwidth.

The remainder of this paper is structured as follows. Section 2 gives system assumptions including the QoS metrics used. Section 3 states the optimal resource allocation problem. Section 4 presents a greedy algorithm that attempts to find the optimal allocation for a given traffic distribution. Section 5 contains some discussion including directions for further research. Section 6 is the conclusion.

II. System Assumptions and QoS Metrics

We are assuming an idealized cellular system with a base station able to support m connections located at the center of each cell. We assume for clarity of exposition that all connections have identical bandwidth requirements. Specifically, we assume each connection made by a mobile user requires one unit of bandwidth (UB) per second. This bandwidth may be to support a voice call, or a data connection with comparable bandwidth requirements. We also assume that system cells are managed in clusters of B cells with a single network call controller being responsible for a cluster. Such an architecture has been proven to be optimal for micro-cellular systems, as it removes the call controller bottleneck that arises due to increased handoff rates in micro- and pico-cellular networks [4].

We consider two QoS metrics, probability of overload, P_O, and expected duration of overload, θ. If we look at a typical system cell, C_0, then P_O is the probability that cell C_0 will be overloaded, having more active connections than it can service. Such an overloaded state can last any length of time, however, θ is the expected length of the overload period. Note that P_O is not the handoff dropping probability. In fact handoff dropping probability is a function of P_O as well as the fraction of connections in the cell in which the overload occurs that must be suspended to remain within base station capacity. The precise calculation of handoff dropping probability is beyond

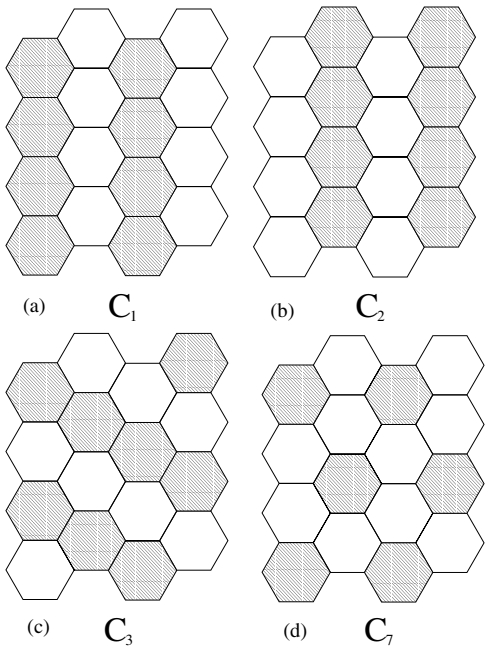

Fig. 1. Four possible coverage patterns.

the scope of the current study. We focus instead on θ, since it will become increasingly important as a system QoS metric as wireless data communications proliferates. After all, data connections such as Web browsing are not dropped when network congestion occurs but must simply wait until the overload passes and resources are available. Wireless data systems must, therefore, take into account the expected duration of such overloads.

In order to guarantee a prescribed level of QoS to subscribing mobiles, the call controller admits at most N connections to the cell cluster. To maintain reasonable QoS levels, N will be strictly less than mB, the full cluster capacity, in order to reserve resources to provide for user mobility. We can then compute P_O and θ as functions of N.

From [4] we have that

$$P_{O,m} = \sum_{i=m+1}^{mB} \frac{\binom{N}{i} (B-1)^{N-i}}{B^N} \tag{1}$$

and

$$\theta = T_{in} \left(\frac{B-1}{N-m} \cdot \frac{P_{O,m}}{P_m} \right) \tag{2}$$

where T_{in} is the mean value of the exponential random variable representing the time a user spends in a cell before handing off [5].

III. PCMA WITH ATTEMPTS TO OPTIMALLY COVER

A. Coverage patterns defined

Even if the network call controller limits the number of connections to some value less than the total cluster capacity, overload conditions can occur. Notice, however, that under the admission control scheme described above, there must exist one or more other system cells experiencing a corresponding "shortage" of connections. Because this shortage represents unused resources, it is plausible that we might be able to shift resources from the under-used cells to the overloaded one or ones. Numerous schemes in the literature have been investigated to do just this [6]. However, they typically assume a circuit-switched model, so there is overhead involved with setting up and releasing circuits. Also, when a channel is borrowed, it is locked out from use by some cells that were previously able to use it. Because PCMA is based on a packet-switched TDMA system, we are able to develop a flexible framework under which resources can be efficiently shifted around among cells only when needed. The framework obviates the need for complex borrowing or locking algorithms. Instead, we are able to instantly make available time slots in the cells in which they are most needed and to then analyze the throughput achievable during these slots.

Because of its use of a capture model, PCMA is ideally suited to support a resource allocation strategy as just described. The model invoked assumes that a packet is successfully received if the ratio of the received signal strength to the sum of all signal strengths from interfering receivers is above a given threshold, b, called the capture ratio. In particular, it is possible to specify an arbitrary set of cells, grant them a set of time slots, and compute the resulting throughput attainable with those slots. In [1] the only case thoroughly considered was uniform allocation. Assuming capture threshold $b = 4$, it was found that the throughput possible is 0.4408, achieved by using two levels of virtual cells, with reuse factors 1 and 3. This uniform allocation will be referred to as coverage pattern C_0.

Of course, by using this same propagation and capture model, we can compute throughputs for non-uniform resource allocation strategies. Some alternative methods of allocation are illustrated in Fig. 1. The shaded cells are given some units of bandwidth, in the form of time slots, for their exclusive use. This set of cells forms a plane coverage pattern.

Two of the possible patterns that can be invoked to cover the plane are illustrated in Fig. 1(a)-(b). Each is a stripe pattern providing for $1/2$ of all system cells to permit user transmission. Fig. 1(c) shows another possible stripe pattern, also covering half of the system cells. This pattern can be viewed as a rotation of the previous one. Furthermore, additional rotations would give three new stripe patterns, C_4 to C_6. Using the interference and capture model from [2] we find each of these patterns provides throughput of 0.7164 to the shaded cells, and zero throughput to the cells not in the pattern. The higher throughput is possible because there is less interference from neighboring cells, since half of them are not transmitting.

Another useful class of coverage patterns are those derived from $1/N$ reuse. Here we shall limit the presentation to those patterns corresponding to $1/3$ reuse, as the reduced interfer-

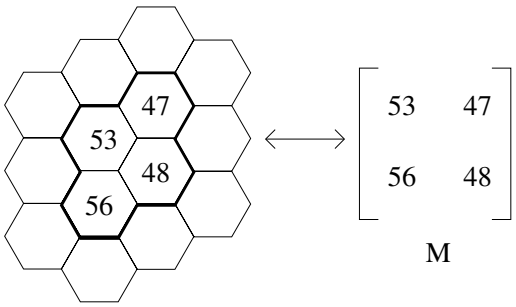

Fig. 2. Mapping used to represent current user distribution state as a matrix.

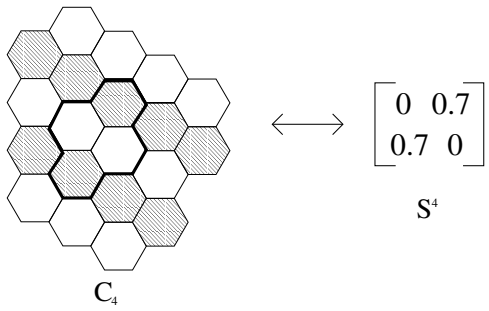

Fig. 3. Mapping from coverage pattern to throughput matrix.

ence of patterns from higher reuse factors is not sufficient to compensate for the reduction in number of cells receiving resources. For a $1/3$ pattern as in Fig. 1(d) there are only one set of virtual cells, of reuse one, since there is very little interference from other transmitting cells due to no neighboring cells simultaneously transmitting. The throughput achieved is 0.9186 for capture threshold $b = 4$. Keep in mind, however, that only $1/3$ of system cells enjoy such high throughput at a given instant. This illustrates the nature of the plane coverage problem. Patterns which provide higher throughput also afford a lower percentage of system cell coverage. In the case of uniform traffic, this tradeoff would render the additional patterns useless, since the uniform pattern would outperform multiple instances of the other patterns. For example, application of three distinct $1/3$ patterns to cover the whole plane results in $0.9186/3 = 0.3062$ of effective throughput, considerably less than the uniform coverage pattern. However, as we shall see, using non-uniform patterns is preferable if they can cover multiple cells having a higher amount of traffic than their neighbors.

B. Formal Problem Specification

We seek a distribution of coverage patterns that will minimize the total number of unserved users in a cell cluster. Before formally stating the problem a mapping is defined from system states onto matrices, and several definitions are given.

First, given an n by n cluster of B cells, we map the current number of connections in each cell to an n by n matrix, M, in the obvious manner. As illustrated in Fig. 2, we view the cells as a collection of rows and columns and let M_{ij} be the demand (e.g., the number of connections requested) in the cell in the ith row and jth column of the cluster. We define the norm of a matrix M as

$$\|M\| = \sum_{i,j} M_{ij}.$$

We shall see that after allocating resources to service demand, the norm of the resulting, reduced matrix is the amount of unserved demand.

A coverage pattern is an allocation of resources to users in particular cells in the cell cluster. Each coverage pattern C_r can also be represented by a matrix. For coverage pattern C_r, let S^r be a matrix of entries where S^r_{ij} is the throughput of cell ij when coverage pattern r is used. The matrix for coverage pattern 4 is shown in Fig. 3. Note that if T UB are allocated to users in cells according to pattern r, then TS^r_{ij} usable UB will be available to users in cell ij.

A cover is a collection of coverage patterns that is used during a particular frame of protocol operation. A cover must indicate how many UB are allocated to each pattern. Therefore, a cover can be specified by a tuple of reals $(n_0, n_1, \ldots)$, where a zero means that particular pattern is unused during the frame. For example, $(3, 2.5, 4.5, 0)$ is a tuple representing allocation among 4 coverage patterns. In this case, 3 UB are used by pattern C_0, while zero UB are allocated to pattern C_3. At present we use arbitrary precision reals to obtain an upper bound on performance. In practice UB are comprised of many time slots which are atomic units. Therefore, in an actual system we have real tuples of precision limited by the reciprocal of the number of time slots per UB. Let a base station under uniform coverage, with throughput matrix S^0, be able to service m users per frame. Then we really have $m/S^0_{ij} = W$ UB to allocate. Note that this holds for any i,j because all entries in S^0_{ij} are equal. Therefore, our cover must be such that no more than W UB are allocated among the various coverage patterns. Specifically, $\sum_i n_i \leq W$.

After applying a cover and reducing the user distribution matrix by the appropriate amount, the norm of the resulting matrix represents the number of unserviced users. Thus, we seek the cover that minimizes the total number of unserved users,

$$\|M - \sum_k n_k S^k\|.$$

IV. Greedy Algorithm

Our first attempt at optimally covering a non-uniform user distribution will be to use a greedy approach. We define a greedy decision strategy as follows. Given a current demand matrix, M^r, define a new matrix by $M^{r\star}_{ij} = 1$ if $M^r_{ij} > 0$ and 0 otherwise. Now, given our current matrix $M^{r\star}$, we do an entry by entry matrix multiply operation with each throughput matrix, S^i. The multiplication is defined for two matrices A and B as

$$(A \otimes B)_{ij} = A_{ij}B_{ij}.$$

We then take the norm of every resulting matrix product. What we have then is a measure of the effectiveness of each coverage pattern at that stage. Contributions are made to the final result for a given cell only if the coverage pattern provides service to that cell and there are still unserved users in the cell.

We ran a simulation using this greedy algorithm. The cell cluster size was $B = 16$ and base station capacity was $m = 50$ UB so that the total cluster capacity was 800. At each stage, we computed $\|M^{r\star} \otimes S^i\|$ and used the coverage pattern i that resulted in the maximum value. This pattern was used to provide one UB of throughput across all cells in the coverage pattern, resulting in a new matrix M^{r+1} of unserved users. The procedure was then repeated until either all users were serviced or the number of UB ran out.

It was immediately evident that the results were not as favorable as expected. A careful examination of a small, four cell cluster reveals that the greedy algorithm is not optimal. We assume for illustration that base station capacity is 10 UB, uniform coverage provides throughput of 0.4, and there are also 6 stripe patterns each providing throughput 0.7. Consequently there are $10/0.4 = 25$ UB to be allocated by each cover.

$$\begin{array}{|c|c|} \hline 7 & 10 \\ \hline 12 & 10 \\ \hline \end{array} \left(\begin{array}{c} \frac{35}{2} \\ \rightarrow \\ C_0 \end{array} \right) \begin{array}{|c|c|} \hline 0 & 3 \\ \hline 5 & 3 \\ \hline \end{array} \left(\begin{array}{c} \frac{30}{7} \\ \rightarrow \\ C_2 \end{array} \right) \begin{array}{|c|c|} \hline 0 & 0 \\ \hline 5 & 0 \\ \hline \end{array} \left(\begin{array}{c} \frac{45}{14} \\ \rightarrow \\ C_4 \end{array} \right) \begin{array}{|c|c|} \hline 0 & 0 \\ \hline 2\frac{3}{4} & 0 \\ \hline \end{array}$$

Here pattern C_0 is used repeatedly until the upper left corner is emptied of users. This requires $(35/2)$ UB and services $(35/2) \times 0.4 = 7$ users per cell. Next pattern C_2 is used because more resources can be directed at the right column. This allocation services $(30/7) \times 0.7 = 3$ users per cell covered. There is then only 1 unserved cell, and pattern C_4 is chosen and covers that cell. Unfortunately, there are only enough UB to service $(45/14) \times 0.7 = 2.25$ of the remaining users, so there are 2.75 unserved users. Compare this, however, to uniform coverage C_0 used completely. In this case we would have served $25 \times 0.4 = 10$ users per cell, resulting in only 2 unserved users.

Given this observation, we tried an alternate approach, termed a *min* algorithm. We ran the greedy algorithm on each resulting user distribution, but then compared the total number of unserved users to those resulting from application of the uniform coverage pattern exclusively. We used the one that resulted in the lower number of unserved users. Results for P_O are reported in Fig. 4. Note that to achieve P_0 of at most 0.01 under uniform allocation, only 581 users can be admitted to the cell cluster. This results in roughly 37% unused capacity considering the full cluster capacity is 800. On the other hand, the min algorithm permits 630 users to be admitted to the cluster while still meeting the same maximum overload probability. This is an increase of over 8% in admission threshold and allows over 22% of the idle capacity to be reclaimed. If the target value for P_O is less than 0.01 the increase in admission threshold is even greater, as can be seen in Fig. 4.

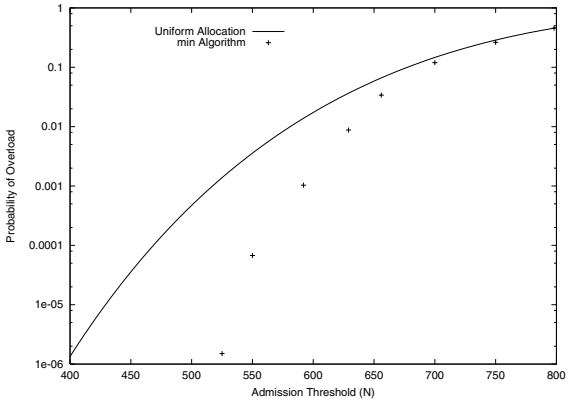

Fig. 4. Probability of overload vs. admission threshold

Results for θ are reported in Fig. 5. Here the normalized expected duration of overload is plotted vs. the admission threshold N. The normalization is the ratio of θ to the expected time a user spends in a cell before handing off, T_{in}. This normalized value remains constant, independent of the handoff rate [4] as can be seen in equation (2). Notice that the results show even greater improvement than for P_0. If we require a maximum normalized θ of 0.04 we could admit only 457 users. However, using the min algorithm permits admission of 760 users, which is a capacity improvement of over 66%. As already indicated, expected duration of overload is quite an important metric, since a user is likely to tolerate a slightly higher overload probability if the overload periods are expected to be quite short. This is especially true for data applications such as Web browsing where delays in pages loading are expected, whether due to an overloaded cell or even non-wireless factors such as high server load. Such delays are quite tolerable if their average length is short. A major benefit of our approach is that lower values of θ are possible than can reasonably be achieved using uniform coverage. Consider if the maximum normalized θ were required to be 0.03. Using the min algorithm we can admit up to 722 users. Under uniform allocation, however, we can admit no more than 324 users, which is not a practical value since it is far less than half of the total cluster capacity. Using the min algorithm, there is less than 10% unused capacity in the cluster.

V. Discussion and Further Research

In a previous section it was mentioned that there is a limit on the precision with which resources can be allocated by the greedy algorithm. We indicate several reasons why this limit is not expected to affect performance significantly. In a system in which user handoffs are infrequent (due to minimal mobility or large cells), many seconds would be likely to pass before any change in distribution had occurred and the greedy algorithm would need to be run again. Given that resources only need be allocated once per user distribution, one particular application of the greedy algorithm would not be allocating 1 UB per user

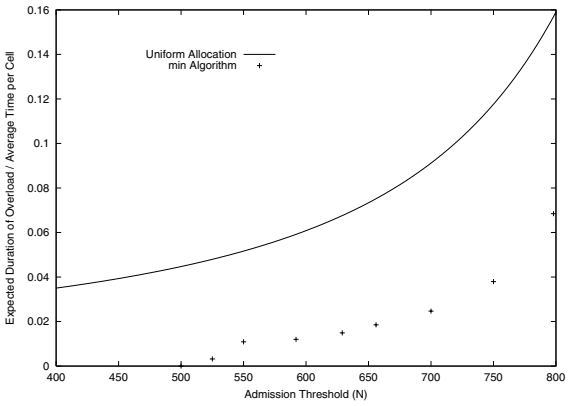

Fig. 5. Normalized expected duration of overload vs. admission threshold.

for one second as was used in the examples for simplicity, but would in fact be allocating many more UB at a time. Since the resources allocated in this case would represent more than one thousand time slots, the precision with which allocation of a set of UB could take place would be at least on the order of thousandths of a UB.

Even if user mobility is rapid, we can make the same arguments if we only re-compute resource allocations with the min algorithm when there has been enough movement to significantly change the user distributions. One user moving into an adjacent cell would not be considered significant. After all, it would likely take several additional users entering the cell to create enough change in loads that recomputing would lead to a better cover and an improvement in performance. This is of course an avenue for further research. We could look at the tradeoff between recomputing often to get a more exact match to current loads, and computing less frequently so that we have more UB to allocate each time and, therefore, finer granularity possible in the allocation. In the latter case we can more closely match the optimal allocation as has been computed assuming arbitrary precision.

In addition to the problem that the greedy algorithm can perform worse than uniform allocation, there is also the fact that different decisions among ties at stages of the greedy algorithm can produce differing results. This may, in fact, indicate the reason the greedy algorithm sometimes performs worse than uniform allocation. Consider the earlier example, but at the second stage use pattern C_6 instead of C_2.

$$\begin{array}{|c|c|} \hline 7 & 10 \\ \hline 12 & 10 \\ \hline \end{array} \left(\begin{array}{c} \frac{35}{2} \\ \rightarrow \\ C_0 \end{array} \right) \begin{array}{|c|c|} \hline 0 & 3 \\ \hline 5 & 3 \\ \hline \end{array} \left(\begin{array}{c} \frac{30}{7} \\ \rightarrow \\ C_6 \end{array} \right) \begin{array}{|c|c|} \hline 0 & 3 \\ \hline 2 & 0 \\ \hline \end{array} \left(\begin{array}{c} \frac{45}{14} \\ \rightarrow \\ C_4 \end{array} \right) \begin{array}{|c|c|} \hline 0 & \frac{3}{4} \\ \hline 0 & 0 \\ \hline \end{array}$$

The total number of unserved users is two less than in the earlier example using the same greedy algorithm. An arbitrary decision was made at step 2 that considerably affects performance. At that stage either pattern C_6 or C_2 yielded the same decision metric of $0.7 \times 2 = 1.4$. So the choice of pattern in such cases has an obvious impact on algorithm performance. Heuristics should be investigated that make such decisions in the case of ties in a manner that minimizes the number of unserved users in the final result of the greedy algorithm. This may solve the problem of occasional inferior performance to uniform allocation and allow us to just use the optimized greedy approach with no min algorithm.

VI. Conclusion

We have investigated the Quality of Service (QoS) performance of Plane Cover Multiple Access under the assumption that the protocol attempts to optimally cover cell clusters based on the current, potentially non-uniform, distribution of users. We have formally specified the covering problem in terms of operations on matrices and have shown that considerable gains in allowable admission threshold are possible while still maintaining a desired QoS in terms of cell overload probability, expected duration of overload, or both. In fact, if expected duration of overload is taken to be the primary QoS metric to be met, as is likely for data applications, lower values are achievable than are even possible under uniform allocation, for practical values of admission threshold. A simulation study was conducted to evaluate the performance of an algorithm that considers a greedy coverage attempt. Given that this algorithm is not optimal, further research involves the search for an optimal algorithm.

References

[1] P. Blair, G.C. Polyzos, and M. Zorzi, "Plane Cover Multiple Access: a new approach to maximizing cellular system capacity," *IEEE Journal on Selected Areas in Communications*, 2001 (in press).

[2] F. Borgonovo, M. Zorzi, L. Fratta, V. Trecordi, and G. Bianchi, "Capture-Division Packet Access for wireless personal communications," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 4, May 1996, pp. 609-622.

[3] P. Blair, G.C. Polyzos, and M. Zorzi, "Plane Cover Multiple Access: a media access control strategy for wireless environments," Proc. International Conference on Universal Personal Communications (ICUPC'98), Florence, Italy, October 1998.

[4] "An architecture and methodology for mobile-executed handoff in cellular ATM networks," *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 8, October 1994, pp. 1365-75.

[5] R. Guerin, "Channel occupancy time distribution in a cellular radio system," *IEEE Transactions on Vehicular Technology*, Vol. 36, No. 3, August 1987.

[6] I. Katzela and M. Naghshineh, "Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey," *IEEE Personal Communications*, Vol. 3, No. 3, June 1996, pp. 10-31.