On QoS Guarantees with Reward Optimization for Servicing Multiple Priority Classes in Wireless Networks

Okan Yilmaz and Ing-Ray Chen
Computer Science Department
Virginia Tech
{oyilmaz, irchen}@vt.edu

***Abstract* –** We report performance characteristics of a class of call admission control (CAC) algorithms designed for servicing multiple priority classes in wireless networks with the goal of quality of service (QoS) satisfaction and reward optimization. By reward, we mean some sort of "value" obtained by the system as a result of servicing multiple priority classes. In this paper we design and evaluate the performance of a new algorithm, *elastic threshold-based CAC*, in terms of the maximum reward obtainable with QoS satisfaction from servicing multiple priority classes with distinct QoS requirements, and compare it with existing partitioning, threshold, and spillover CAC algorithms. We also develop a heuristic-based search method to determine the best threshold-value sets used for multiple service classes by sequentially adjusting these thresholds based on the total reward and rejection rate vs. QoS constraints of each service class. We demonstrate through test cases and simulation that elastic threshold-based CAC outperforms existing CAC algorithms for QoS satisfaction and reward optimization in solution optimality for serving multiple QoS service classes in wireless networks.

Index Terms – Quality of service, call admission control, performance analysis, reward optimization, wireless networks.

I. Introduction

Next generation mobile wireless networks will carry real-time multimedia services such as video and audio and non-real-time services such as text and image. An important goal is to adapt to growing user demand without compromising the Quality of Service (QoS) requirements.

While QoS for service classes can cover a wide variety of performance metrics, two important QoS metrics in cellular wireless networks are the blocking probability of new calls and the dropping probability of handoff calls due to unavailability of wireless channels. A handoff occurs when a mobile user with an ongoing connection leaves the current cell and enters another cell. An ongoing connection may be dropped during a handoff if there is insufficient bandwidth in the new cell to support it. We can reduce the handoff call drop probability by rejecting new connection requests. However, reducing the handoff call drop probability could result in an increase in the new call blocking probability.

Call admission control (CAC) for single-class network traffic, such as voice (real-time), has been studied extensively in the literature [1-10, 16]. There have been CAC algorithms proposed that partition system resources and allocate distinct partitioned resources to serve distinct service classes [11-18]. In particular, Ogbonmwan, Li and Kazakos [11] investigated the use of thresholds for a system with two service classes. Three separate thresholds are used to reserve channels for voice handoff calls, new voice calls, and data handoff calls. These threshold values are periodically reevaluated based on workload conditions. Haung and Ho [12] presented a distributed CAC algorithm that runs in each cell and partitions channel resources in the cell into three partitions: one for real-time calls, one for non-real-time calls, and one for both real-time and non-real-time calls to share. To be able to satisfy more stringent QoS requirements of handoff calls, they also applied a threshold value to new calls. To estimate call arrival rates to each cell in the heterogeneous network, they used an iterative algorithm. Ye, Hou and Papavassilliou [14] proposed a bandwidth reservation and reconfiguration mechanism to facilitate handoff processes for multiple services.

All these CAC algorithms cited above make acceptance decisions for new and handoff calls to satisfy QoS requirements in order to keep the dropping probability of handoff calls and/or the blocking probability of new calls lower than pre-specified thresholds. Also all these algorithms concern QoS requirements, without considering "value" issues associated with service classes, i.e., what value priority service classes will bring to the system. Chen et al. [15] first proposed the concept of maximizing the "reward" of the system through CAC in the context of multimedia services. They subsequently studied spillover and other CAC algorithms [17, 18]. In this paper we propose a new threshold-based CAC algorithm, *elastic threshold-based CAC*, with the goal to maximize the system reward rate with QoS guarantees. Here we note that "reward" is a generic term, referring to some sort of "value" brought to the system due to services. It could map to "revenue" (monetary value) from the service provider's perspective. Our results show that elastic threshold-based CAC developed is capable of satisfying QoS while generating higher rewards compared with existing CACs.

The rest of the paper is organized as follows. Section 2 states the system model and gives assumptions used in characterizing the operational environment of a mobile wireless network. Section 3 describes the elastic threshold-based CAC algorithm for reward optimization with QoS guarantees in mobile wireless networks. Section 4 analyzes performance characteristics of elastic threshold-based algorithm and compares its performance against existing CAC algorithms in terms of maximum reward generated with QoS guarantees. Finally, Section 5 summarizes the paper and outlines some future research areas.

II. System Model

We consider that a wireless cellular network consists of a number of cells, each of which has a base station to provide network services to mobile hosts within the cell. We assume that distinct service classes exist, each characterized by its service type. For example, the service type can be *real-time* and *non-real time*. Further, there are *handoff* and *new* calls for each service type, with handoff calls having a higher priority than new calls since disconnection of an ongoing call is considered very undesirable from the user perspective. Each service type, other than requiring a number of bandwidth channels to satisfy its intrinsic bandwidth QoS requirement, imposes a system-wide QoS requirement. For example, the handoff call drop probability of a service type may be less than 5% as a QoS requirement. Dropped handoff calls dissatisfy users more than blocked new calls do, so the drop probability requirement of handoff calls is likely to be more stringent than the blocking probability requirement of new calls. In general, assume that service class *i* has QoS threshold requirements in the handoff blocking probability Bt^i_h and the new call blocking probability Bt^i_n with $Bt^i_n > Bt^i_h$.

From the perspective of a single cell, each service class is characterized by its call arrival rate (calls per unit time), including new calls initiated by mobile users in the cell and for handoff calls from neighbor cells, and its departure rate (calls completed per unit time). Let λ^i_n denote the arrival rate of *new* calls of service class *i* and μ^i_n be the corresponding departure rate. Similarly, let λ^i_h denote the arrival rate of *handoff* calls of service class *i*, and μ^i_h be the corresponding departure rate. These parameters can be determined by inspecting statistics collected by the base station in the cell and by consulting with base stations of neighbor cells [17].

Without loss of generality we assume that a cell has *C* channels where *C* can vary depending on the amount of bandwidth available in the cell. When a call of service class *i* enters a handoff area from a neighboring cell, a handoff call request is generated. Each call has its specific QoS bandwidth requirement dictated by its service traffic type attribute. Let k^i denote the number of channels required by a service call of service class *i,* e.g., 4 for multimedia and 1 for voice calls.

One way to related "reward" with "value" earned by the system is to consider the amount of revenue earned by the service provider. While many algorithms exist [15], the most prevalent with general public acceptance to date is the "charge-by-time" scheme by which a user is charged by the amount of time in service. Let v^i be the reward for a call of service class *i* per unit time. That is, if a call of service class *i* is admitted into a cell, and subsequently handed off to the next cell or terminated in the cell, a reward of v^i multiplied with the amount of time the service is rendered in the cell will be "earned" by the system. There is no distinction for handoff vs. new calls in the amount of reward earned as long as the call is in the same service class. The performance model developed in the paper allows the service provider to calculate the reward earned *per unit time* under a CAC algorithm by each individual cell such that the reward obtained by the system is maximized while satisfying QoS constraints.

For ease of disposition, we will consider two priority service classes in this paper. The CAC algorithm developed can be easily generalized for multiple service classes. The total reward R_T generated by each cell per unit time would be the sum of the rewards generated by various priority classes:

$$R_T = R^1_h + R^1_n + R^2_h + R^2_n \qquad (1)$$

Here R^i_h represents reward earned from servicing class *i* handoff calls per unit time, and R^i_n represents reward earned from servicing class *i* new calls per unit time.

The QoS constraints are expressed in terms of blocking probability thresholds, Bt^1_h, Bt^1_n, Bt^2_h, and Bt^2_n, for class 1 handoff calls, class 1 new calls, class 2 handoff calls, and class 2 new calls, respectively. Suppose that the *observed* handoff dropping probability and new call blocking probability of class *i* generated by a CAC algorithm are B^i_h and B^i_n. Then the imposed QoS constraints are considered satisfied when:

$$B^1_h < Bt^1_h;\ B^1_n < Bt^1_n;\ B^2_h < Bt^2_h;\ B^2_n < Bt^2_n. \qquad (2)$$

III. Elastic Threshold-Based Call Admission Control

Our elastic threshold-based CAC algorithm derives from a threshold-based CAC developed [17] which applies a separate and distinct threshold to each service type. A threshold-based CAC algorithm essentially shares all channels available among all service classes to establish a higher utilization. It applies thresholds to limit the traffic from low-priority calls and thus reserves more bandwidth for high-priority calls. Although this algorithm is quite successful, it suffers from the use of discrete thresholds which cut traffic from service classes abruptly and reject any further traffic. We develop elastic threshold-based CAC to improve the effectiveness of threshold-based CAC in terms of QoS satisfaction and thus higher rewards generated by applying elastic thresholds.

For ease of disposition, we assume that two service classes exist with class 1 being the high-priority class. The algorithm can be easily extended to the case in which more classes exist.

Our elastic threshold-based CAC algorithm uses two thresholds, a high (*Hth)* threshold and a low (*Lth)* threshold, for each service class in the system. The CAC algorithm rejects a fraction of class *i* new calls when LTh^i_n is reached and rejects all class *i* new calls when HTh^i_n is reached. Similarly, the CAC algorithm starts blocking a fraction of class *i* handoff calls when LTh^i_h is reached and blocks all class *i* handoff calls when HTh^i_h is passed. For each threshold value, elastic threshold-based CAC assumes that the threshold is reached if accepting an incoming call will cause the number of channels used exceeds the threshold value. The probabilities of accepting a new call and a handoff call of service class *i*, represented by P^i_n and P^i_h, are given by:

$$P_n^i = \begin{cases} 1 & n+k^i \le LTh_n^i \\ 1-\left[\left(n+k^i-LTh_n^i\right)\Big/\left(C-LTh_n^i\right)\right] & LTh_n^i < n+k^i \le HTh_n^i \\ 0 & HTh_n^i < n+k^i \end{cases} \qquad (3)$$

$$P_h^i = \begin{cases} 1 & n+k^i \le LTh_h^i \\ 1-\left[\left(n+k^i-LTh_h^i\right)\Big/\left(C-LTh_h^i\right)\right] & LTh_h^i < n+k^i \le HTh_h^i \\ 0 & HTh_h^i < n+k^i \end{cases} \qquad (4)$$

Here *C* and *n* represent the total number of channels and the number of channels that have been allocated in the system, respectively. Figure 1 illustrates the thresholds and call arrivals in elastic threshold-based CAC running in a system with *C* channels. Each color represents the call arrival rate of a different service call type accepted by the system depending on the total number of channels used by all service types. For example, LTh^2_n (a low threshold) is triggered if a new low-priority class 2 call arrives when the number of channels used by the system is greater than $LTh^2_n - k^2$. After this, the CAC starts rejecting a fraction of class 2 call arrivals until a class 2 new call arrives causing the number of channels used greater than $HTh^2_n - k^2$. After the high threshold of class 2 class is reached, the system stops accepting class 2 new calls.

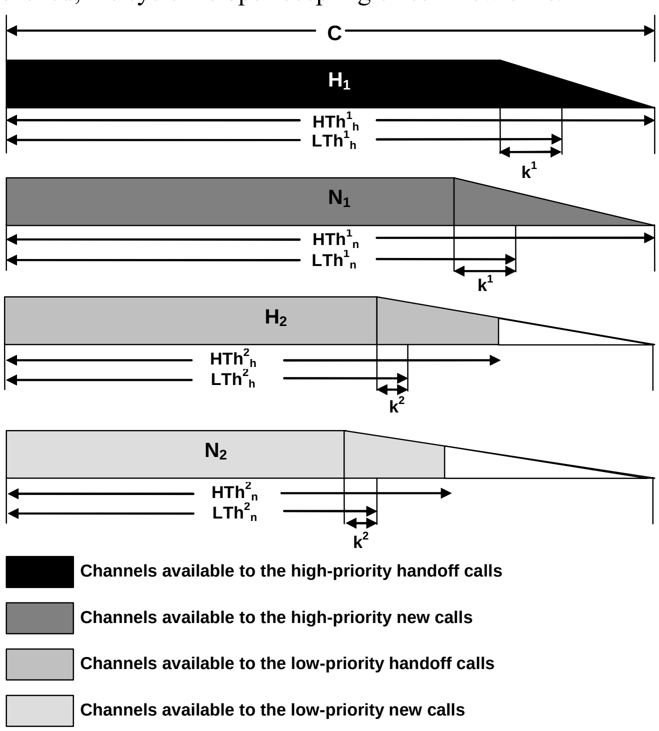

Figure 1: Elastic Threshold-Based Admission Control.

By allowing multiple service call types to share channels and by limiting call arrivals of low-priority service classes by using a pair of best high and low threshold values, the elastic threshold-based algorithm produces optimal solutions. To satisfy the imposed QoS constraints, the CAC looks for "legitimate" low and high thresholds in the form of (LTh^1_h, LTh^1_n, LTh^2_h, LTh^2_n) and (HTh^1_h, HTh^1_n, HTh^2_h, HTh^2_n) generated such that B^1_n, B^1_h, B^2_n, and B^2_h satisfy the QoS constraints specified by Condition (2).

As the system services distinct class types with distinct channel demands, it requires the use of a sophisticated model to evaluate its performance. We develop a mathematic model based on Stochastic Petri Net (SPN) as shown in Figure 2. We use *places* (circles) to hold calls being admitted and serviced by the system. We use *transitions* (vertical bars) to model event arrivals. A transition is enabled when its input place (if exists) is non-empty and its associated enabling predicate (if exists) is evaluated *true*. A transition rate is associated with a transition to specify how often the event associated with the transition occurs. In Figure 2, place UC^i_n holds service class *i* new calls, with $M(UC^i_n)$ representing the number of new calls it holds. Place UC^i_h holds service class *i* handoff calls with $M(UC^i_h)$ representing the number of handoff calls it holds. Transition E^1_h models the arrival of a class 1 handoff call at rate $f(\lambda^1_h)$; E^1_n models the arrival of a class 1 new call at rate $f(\lambda^1_n)$; E^2_h models the arrival of a class 2 handoff call at rate $f(\lambda^1_h)$; E^2_n models the arrival of a class 2 new call at rate $f(\lambda^2_n)$. At the other end, S^i_h models the departures of a class *i* handoff call with a service rate of $M(UC^i_h)$ multiplied with per-call service rate μ^i_h; and S^i_n model the departure of a class *i* new call with a service rate of $M(UC^i_n)$ multiplied with per-call service rate μ^i_n.

A service request arrival is rejected if upon acceptance of the service request, the number of channels used in the system exceeds the high threshold (HTh^i_h for class *i* handoff and HTh^i_n for class *i* new calls) for the service call type. Therefore, we assign enabling predicates to guard E^1_h, E^1_n, E^2_h, and E^2_n with HTh^1_h, HTh^1_n, HTh^2_h, and HTh^2_n being the constraints. Specifically, the enabling predicate of E^i_h, E^i_n, E^1_h, and E^1_n are:

$$\left[M\left(UC_h^1\right)+M\left(UC_n^1\right)\right]k^1+\left[M\left(UC_h^2\right)+M\left(UC_n^2\right)\right]k^2+k^i \le HTh_h^i \qquad (5)$$

$$\left[M\left(UC_h^1\right)+M\left(UC_n^1\right)\right]k^1+\left[M\left(UC_h^2\right)+M\left(UC_n^2\right)\right]k^2+k^i \le HTh_n^i \qquad (6)$$

The arrival rates of class *i* new calls and class *i* handoff calls are $f\left(\lambda_h^i\right)=\lambda_h^i$ and $f\left(\lambda_n^i\right)=\lambda_n^i$ if the following constraints are satisfied respectively:

$$\left[M\left(UC_h^1\right)+M\left(UC_n^1\right)\right]k^1+\left[M\left(UC_h^2\right)+M\left(UC_n^2\right)\right]k^2+k^i \le LTh_h^i \qquad (7)$$

$$\left[M\left(UC_h^1\right)+M\left(UC_n^1\right)\right]k^1+\left[M\left(UC_h^2\right)+M\left(UC_n^2\right)\right]k^2+k^i \le LTh_n^i \qquad (8)$$

On the other hand, the arrival rates of class *i* new calls and class *i* handoff calls are:

$$f\left(\lambda_h^i\right)=\left(\frac{C-\left(\left[M\left(UC_h^1\right)+M\left(UC_n^1\right)\right]k^1+\left[M\left(UC_h^2\right)+M\left(UC_n^2\right)\right]k^2+k^i\right)}{C-LTh_h^i}\right)\lambda_h^i \qquad (9)$$

$$f\left(\lambda_n^i\right)=\left(\frac{C-\left(\left[M\left(UC_h^1\right)+M\left(UC_n^1\right)\right]k^1+\left[M\left(UC_h^2\right)+M\left(UC_n^2\right)\right]k^2+k^i\right)}{C-LTh_n^i}\right)\lambda_n^i \qquad (10)$$

if the following constraints are satisfied respectively:

$$LTh_h^i \le \left[M\left(UC_h^1\right)+M\left(UC_n^1\right)\right]k^1+\left[M\left(UC_h^2\right)+M\left(UC_n^2\right)\right]k^2+k^i \le HTh_h^i \quad (11)$$

$$LTh_n^i \le \left[M\left(UC_h^1\right)+M\left(UC_n^1\right)\right]k^1+\left[M\left(UC_h^2\right)+M\left(UC_n^2\right)\right]k^2+k^i \le HTh_n^i \quad (12)$$

Finally, $f\left(\lambda_h^i\right)=0$ and $f\left(\lambda_n^i\right)=0$ if E_n^i and E_h^i are enabled. Consequently, it follows that the blocking/dropping probabilities B_n^1, B_h^1, B_n^2, and B_h^2 can be obtained by:

$$B_h^i=\frac{\lambda_h^i-rate\left(f\left(\lambda_h^i\right)\right)}{\lambda_h^i} \quad (13) \qquad B_n^i=\frac{\lambda_n^i-rate\left(f\left(\lambda_n^i\right)\right)}{\lambda_n^i} \quad (14)$$

Here $rate\left(f\left(\lambda_h^i\right)\right)$ and $rate\left(f\left(\lambda_n^i\right)\right)$ are obtained as the expected value of a random variable X defined as $X=f\left(\lambda_h^i\right)$ and $X=f\left(\lambda_n^i\right)$ if E_h^i and E_n^i are disabled, and zero otherwise. The reward earned per unit time, R_h^i, and R_n^i in (1), can be calculated as:

$$R_h^i=\left(1-B_h^i\right)\lambda_h^i\cdot v^i \quad (15) \qquad R_n^i=\left(1-B_n^i\right)\lambda_n^i\cdot v^i \quad (16)$$

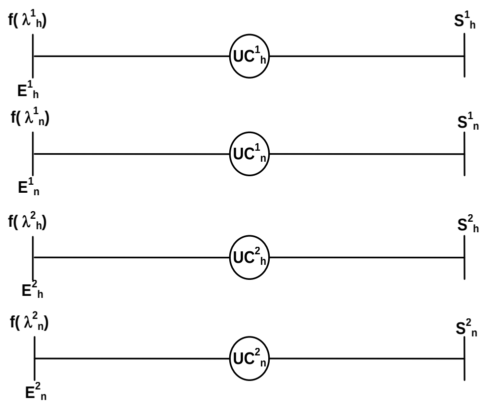

Figure 2: SPN Model for Elastic Threshold CAC.

3.1 Search for Optimal Threshold Values that Generate Maximal Reward with QoS Guarantees

We develop a heuristic-based search algorithm to determine the optimal threshold combinations that would maximize the reward earned per unit time while satisfying QoS constraints specified in (2). The method essentially is a greedy search method to find a near optimal solution quickly at the expense of search optimality. However, as we shall see, the solution found by this algorithm is at least as good as or better than the solutions found by base algorithms.

The elastic threshold-based CAC algorithm utilizes a greedy search method to determine a legitimate solution which maximizes the reward rate. This algorithm determines the optimal threshold combination in two steps:

Step 1: finding a legitimate solution;
Step 2: determining a locally optimal solution by applying a greedy search starting from the legitimate solution found in the first step.

Our elastic threshold-based CAC algorithm accepts a delta (Δ) value as input. This value is used to determine the set of threshold combinations in the range of current threshold $\pm\ \Delta$. The optimality search stops when the current optimal reward is higher than the reward generated for all threshold combinations in this set.

In the first step we try two independent methods to determine a legitimate solution which satisfies the QoS constraints of all service call types. In the first method, we set all thresholds to *C* and check if this combination is legitimate. We reduce the bandwidth used by low-priority classes by lowering the low threshold of low-priority classes until we find a legitimate solution or we start missing QoS constraints of low-priority classes. When we cannot limit the arrivals of low-priority classes anymore we start reducing arrivals of high-priority new calls as well by lowering their low threshold. If none of these approaches generates a legitimate solution, this method returns no legitimate solution.

The second method is called only if the first method fails to find a legitimate solution. In this method we first determine the minimum number of channels needed to satisfy the QoS constraint of each service call type. This can be done by modeling each service call type as an M/M/n/n queue and determining the minimum number of channels that would satisfy the QoS constraints. This helps eliminate all threshold combinations that do not provide at least the minimum number of channels to each service call type in the search. In subsequent iterations, we increase the low thresholds of low-priority calls if the low threshold is less than the high threshold; otherwise, we increase the high threshold until the high threshold reaches *C* or we find a legitimate solution. Similarly, we increase the low threshold of high-priority new calls to satisfy the QoS constraints of this service call type. If we cannot satisfy the QoS constraints of high-priority handoff calls, we decrease the high thresholds of low-priority calls and the high threshold of high-priority new calls. This step is done to rollback to a previous state when these new thresholds cannot find a solution. At any point if we determine that we have a threshold combination that has been evaluated, we break looping and continue with checking threshold values one less or one more than the current threshold value. This last check is done because we may encounter previously evaluated thresholds for borderline cases and retrying the same threshold would put us in an infinite loop.

After finding a legitimate solution, we attempt adjacent threshold values in the range of current threshold $\pm\ \Delta$ to determine a legitimate solution with a higher reward rate. When no adjacent threshold returns a higher reward rate, the CAC algorithm returns the optimal value.

IV. Numeric Data and Analysis

We compare elastic threshold-based CAC with existing CAC algorithms including partitioning, threshold-based [17], and spillover [18] CAC algorithms in terms of reward maximization with QoS guarantees. To compare these algorithms, we use a simulation system and collect data to measure arrival and departure rates by using the workload estimation method in [17]. Admission decisions are made by using CACs.

4.1 Performance Comparison in Solution Efficiency

We use a simulated system with a wraparound architecture with each cell having 6 adjacent cells as shown on Figure 3. We configure the system with 1024 mobile users roaming among these cells via the popular random waypoint model [18] by which each mobile user has a random speed and a random destination during travel, and a random pause time when reaching the destination. After reaching the destination and pausing for a random time, a user continues moving to a new destination with a new speed. We use Poisson distribution to model call arrivals and exponential distribution to model the duration of a call. We randomly assign call inter-arrival and call departure rates to each mobile user by using uniform distribution and we test various traffic conditions by changing the number of mobile units in the system. Table 1 summarizes the system parameters used in the simulation. We calculate the workload conditions, based on which CACs are used to determine the optimal channel and threshold allocations which would maximize the system reward obtainable while satisfying QoS constraints of priority service calls.

Table 1: Parameters used in Simulation Study.

Parameter	Description	Value
C	Number of channels in a cell	80
Bt^1_h	QoS constraint for class 1 handoff calls	0.02
Bt^2_h	QoS constraint for class 2 handoff calls	0.04
Bt^1_n	QoS constraint for class 1 new calls	0.05
Bt^2_n	QoS constraint for class 2 new calls	0.1
k^1	Number of channels required by a call of service class 1 (e.g., video)	4
k^2	Number of channels required by a call of service class 2 (e.g., audio)	1
v^1	Reward for a call of service class 1 per unit time	80
v^2	Reward for a call of service class 2 per unit time	6

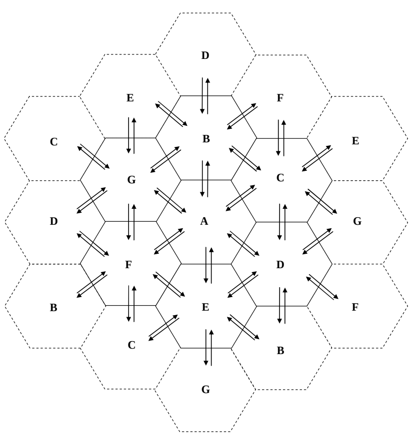

Figure 3: A Simulated Wireless Cellular Network with a Wrap-Around Structure.

Figure 4 shows the reward generation rate versus the number of mobile units in the system. We see partitioning CAC performs the worst in terms of reward rate among the four algorithms evaluated. In our simulation partitioning CAC could not generate legitimate solutions after we increase the number of mobile units in the system to 563. Threshold-based and spillover CAC algorithms continue generating legitimate solutions up to 768 mobile users. After this point these two algorithms also fail to generate legitimate solutions. Finally, elastic threshold-based CAC is able to generate solutions for up to 819 mobile users, after which none of the CAC algorithms is able to satisfy QoS requirements of all service call types. The analytical results obtained from evaluating the SPN model are also shown in Figure 4, which are validated by simulation results.

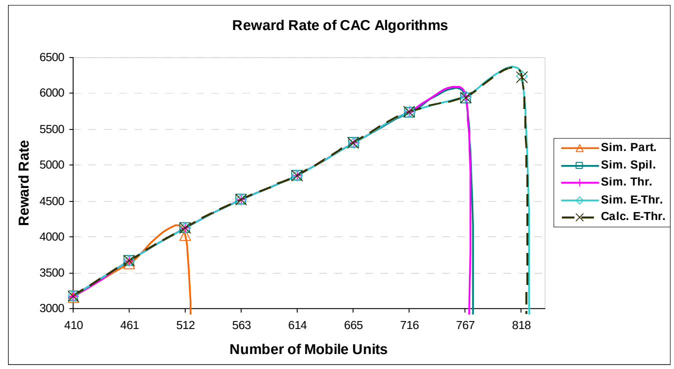

Figure 4: Reward Rate versus Number of Mobile Units.

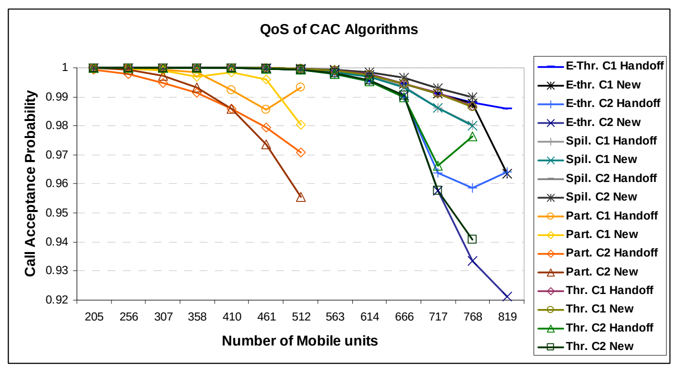

Figure 5: QoS of Admission Control Algorithms.

Figure 5 compares call acceptance ratio obtained by various CAC algorithms ordered by service call type. Partitioning CAC (left of Figure 5) rejects about 1% of low-priority calls when the number of mobile users is 358. When the number of mobile users reaches 512, it rejects 0.7%, 2%, 3%, and 4.5% of class 1 handoff calls, class 1 new calls, class 2 handoff calls, and class 2 new calls, respectively. After this point partitioning CAC could not handle higher traffic with QoS guarantees. This algorithm could not perform as good as other algorithms because it partitions channel resources and serves only a single service call type in each partition. By disallowing channel sharing, partitioning CAC performs significantly worse than other CAC algorithms.

We observe that threshold-based CAC and spillover CAC have similar performance characteristics. Both leverage multiplexing power through channel sharing to improve acceptance ratio. Spillover CAC is able to handle higher traffic by reserving more channels to the partitions allocated to high-priority service classes. On the other hand, threshold-based CAC is able to limit the bandwidth used by low-priority service classes by setting low thresholds. When we increase the number of mobile users to 819, only elastic threshold-based CAC is able to satisfy the high QoS requirement of class 1 handoff calls. As shown in Figure 5, elastic CAC is able to limit the acceptance rate of low-priority new calls to 92% by properly adjusting the low and high threshold values of low-priority new calls. Similarly, it accepts about 96.5% of class 1 new calls and class 2 handoff calls by applying elastic thresholds to these service call types. Although the call acceptance rate of class 1 handoff calls is low, the QoS constraint of 2% level is well satisfied. Overall we observe that elastic threshold-based CAC is able to serve higher traffic with QoS guarantees and generate higher rewards by increasing the bandwidth resources available for high-priority calls, by properly adjusting the pair of thresholds associated with each service type.

V. Conclusions and Future Work

In this paper we discussed design concepts of CAC for both QoS satisfaction and reward optimization and developed and analyzed elastic threshold-based CAC for serving multiple service classes in mobile wireless networks. We compared the performance of elastic threshold-based CAC with existing CAC algorithms in solution optimality. We presented a greedy search algorithm that determines a near optimal solution for elastic threshold-based CAC. We discovered that partitioning CAC performs poorly among all. Threshold-based and spillover CAC algorithms perform reasonably well when the mobile user population is moderate. However, as the user population increases and thus the aggregate traffic increases, they fail to generate a legitimate solution to satisfy QoS constraints of service classes. We demonstrated that the elastic threshold-based CAC algorithm developed even in heavy load conditions is still capable of satisfying QoS requirements and could continue to generate high rewards despite increased traffic generated by high population. We attribute the superiority of elastic threshold-based CAC to its elastic threshold functionality capable of leveraging the low threshold to regulate traffic (rejecting just a fraction of traffic) and the high threshold to reject traffic generated by service calls.

In the future, we plan to analyze the worst case time complexity and computational requirement and more thoroughly evaluate the performance of elastic threshold-based CAC through mobility trace data combined with test cases generated by a random test case generator.

References

[1] Y. B. Lin and I. Chlamtac, *Wireless and Mobile Network Architecture*, John Wiley and Sons, 2001.

[2] D. Hong and S. S. Rappaport, "Priority oriented channel access for cellular systems serving vehicular and portable radio telephones", *Communications, Speech and Vision, IEE Proceedings I.* Vol. 131, No. 5, Oct 1989, pp. 339-346.

[3] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and non-prioritized handoff procedures", *IEEE Transactions on Vehicular Technology*, Vol. VT35. No.3, August 1986.

[4] R. Guerin, "Queuing-blocking systems with two arrival streams and guarded channels", *IEEE Transactions on Communication*, Vol. 36, February 1988, pp. 153-163.

[5] B. Epstein and M. Schwartz, "Reservation strategies for multi-media traffic in a wireless environment," *45th IEEE Vehicular Technology Conference*, 1995, pp. 165-169.

[6] Y. Fang, "Thinning algorithms for call admission control in wireless networks", *IEEE Transactions on Computers*, Vol. 52, No. 5, May 2003, pp. 685-687.

[7] J. Wang, Q. Zeng and D.P. Agrawal, "Performance analysis of a preemptive and priority reservation handoff algorithm for integrated service-based wireless mobile networks", *IEEE Transactions on Mobile Computing*, Vol. 2, No. 1, January-March 2003, pp. 65-75.

[8] N. Nasser and H. Hassanein, "Prioritized multi-class adaptive framework for multimedia wireless networks," *IEEE International Conference on Communications*, Paris, France, June 2004, pp. 4295-4300.

[9] Y. Zhang and D. Liu, "An adaptive algorithm for call admission control in wireless networks," *IEEE Global Telecommunications Conference*, San Antonio, TX, November 2001, pp.3628-3632.

[10] H. Chen, S. Kumar, and C.-C. J. Kuo, "Differentiated QoS aware priority handoff in cell-based multimedia wireless network," *IS&T/SPIE's 12th International Symposium*, San Joes, CA, January 2000, pp. 940-948.

[11] S. E. Ogbonmwan, W. Li and D. Kazakos. "Multi-threshold bandwidth reservation scheme of an integrated voice/data wireless network," *2005 International Conference on Wireless Networks*, Communications and Mobile Computing, Maui, Hawaii, June 2005, pp. 226-231.

[12] Y.-R. Haung and J.-M. Ho, "Distributed call admission control for a heterogeneous PCS network," *IEEE Transactions on Computers*, 51, 2002. pp. 1400-1409.

[13] B. Li, C. Lin, and S.T. Chanson, "Analysis of a hybrid cutoff priority algorithm for multiple classes of traffic in multimedia wireless networks", *Wireless Networks*, Vol. 4, 1998, pp. 279-290.

[14] J. Ye, J. Hou, and S. Papavassilliou, "A comprehensive resource management for next generation wireless networks", *IEEE Transactions on Mobile Computing*, Vol. 1, No. 4, 2002, pp. 249-263.

[15] I.R. Chen and C.M. Chen, "Threshold-based admission control policies for multimedia servers," *The Computer Journal*, Vol. 39, No. 9, 1996, pp. 757-766.

[16] L. Huang, S. Kumar, and C.-C. J. Kuo, "Adaptive resource allocation for multimedia QoS management in wireless networks," *IEEE Transactions on Vehicular Technology*, Vol. 53, 2004, pp. 547-558.

[17] I.R. Chen, O. Yilmaz and I.L. Yen, "Admission control algorithms for revenue optimization with QoS guarantees in mobile wireless networks," *Wireless Personal Communications*, Vol. 38, No. 3, August 2006, pp. 357-376.

[18] O. Yilmaz, I.-R. Chen, G. Kulczycki, and B. Frakes "Spillover-partitioning call admission control for revenue optimization with QoS guarantees for multiple service classes in mobile wireless networks," *3rd International Workshop on Performance Analysis and Enhancement of Wireless Networks,* Okinawa, Japan, March 2008.

[19] D.B. Johnson and D.A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, 1996, pp. 153-181.