

On Detecting Unsteady Demand in Mobile Networking Environment *

V.V. Shakhov, H. Choo, and H.Y. Youn

School of Electrical and Computer Engineering
Sungkyunkwan University, Suwon 440-746, KOREA
{vova, choo, youn}@ece.skku.ac.kr

Abstract. One of the key issues in mobile communication system is how to predict the number of calls per each cell. It is an important parameter and usually assumed as random Poisson value. For effective management of cellular network, the average number of calls should be traced and the changes in the numbers need to be promptly detected. In this paper we propose an algorithm detecting the changes in the behavior of the users using the technique proposed for point-of-change problem based only on the number of call arrivals. Computer simulation reveals that the proposed method can effectively detect the discord, and the developed model is very accurate as showing mostly less than 1% differences.

1 Introduction

Recently information technology has been evolving into the direction leading to the convenience of users. Mobile and wireless communication have a significant role in this IT networking era due to the request of users. One of main elements of mobile networking, especially for network design, is the prediction mechanism for the number of calls per unit time in each cell [1]. The number is related to the number of frequency channels, the call blocking probability, and many other key issues of resource management in mobile computing. With steady-state behavior of mobile users, several techniques [1] can be employed for effective mobility management.

If the number of mobile users is simply modeled as a constant with a certain level of blocking probability, the required number of channels can be easily obtained. However, as we know, behavior of mobile users and their movement change dynamically. More specifically, the average and maximum number of calls can widely fluctuate. If they increase, the call blocking probability also increases and thus the quality of service drops. Meanwhile, if they decrease, less amount of resource will be required for maintaining the same quality of service. Hence, the number of calls is an important parameter in mobile network. For effective

* This work was supported in part by Brain Korea 21 and grant No. 2000-2-30300-004-3 from the Basic Research Program of Korea Science and Engineering Foundation. Dr. Choo is the corresponding author.

management of cellular network, the average number of calls should be traced and the changes in the numbers need to be promptly detected.

Sometimes it is enough to use a simple deterministic model for calculating the number of calls, where the parameters of the model are known. Usually a flow of call arrivals is represented by a stochastic model, and the number of calls is a random value decided depending on the behavior of users.

Several models for analyzing the movement of users were proposed in [2, 3]. The models require some additional information such as the speed of mobile users, the direction of movements, travel path, and so on. Since a lot of factors affect the system performance, an approximation technique is employed for simplifying the modeling. For example, assumption of uniform distribution for the speed of mobile users, a cellular network as a set of identical hexagons, etc. Note, however, that movement of users is irregular, and the assumptions employed for the sake of simplicity can cause unrealistic modeling for the average number of calls. In this paper, therefore, it is offered to detect the change in the behavior of the users using the point-of-change problem solution technique based only on the number of call arrivals.

The rest of the paper is organized as follows. Section 2 provides the basic notation and previous works. The main algorithm of discord detection under the Poisson distribution and an admissible lag is presented in Section 3. In Section 4, we present the results of numerous simulation experiments with the proposed algorithm. Section 5 is a brief conclusion.

2 Preliminaries

To calculate the required number of frequency channels or traffic load in a cell, we have to know the number of calls per unit time which is a random number. Let N^c be the required number of calls, N be the total number of users in a cell, L be the probability of a user to have a mobile terminal, and k be the probability of mobile users making network access. Then $N^c = NLk$. Here, N depends on the average distance between mobile users and k depends on the speed of mobile users [1].

Actually, the larger the sojourn time for a mobile user is, the higher the network access probability is. If traffic congestion occurs, the average distance between adjacent mobile terminals decreases, and thus the sojourn time increases. Hence, both N and N^c increase. In this situation, more channels are required for customers to have a certain level of quality of service. Refer to Figure 1. In this figure, N_i , k_i and N_i^c are the parameters discussed above before ($i = 1$) and after ($i = 2$) traffic congestion. Here we assume that each user carries a mobile terminal ($L = 1$) and two channels are assigned to a cell.

The number of calls can be predicted by mathematical modeling. In this section some models proposed for modeling the behavior of the users are discussed. Their approaches are usually based on a certain regularity of user movement. The models proposed in [2, 4] assume that the average speed of users and their directions are same in each cell. However, movements of customers are usually

different in practice, and even the information on the direction and speed of the users are unavailable. In [5], the authors consider irregular movements of mobile users. For predicting the future location of mobile users a Markov chain model and database of the movement patterns are used. This scheme does not work if preliminary information such as pattern database is either absent or obsolete. On the other hand, an approach for predicting the behavior of the users based only on current location information is considered in [6]. This scheme is costly because no previous information is used while highly accurate data observation is necessary. A mobility prediction scheme based on neuro-fuzzy theory is offered in [3]. Here a unique reason can affect the behavior of a user, and thus all previous information and regulations can be out-of-date.

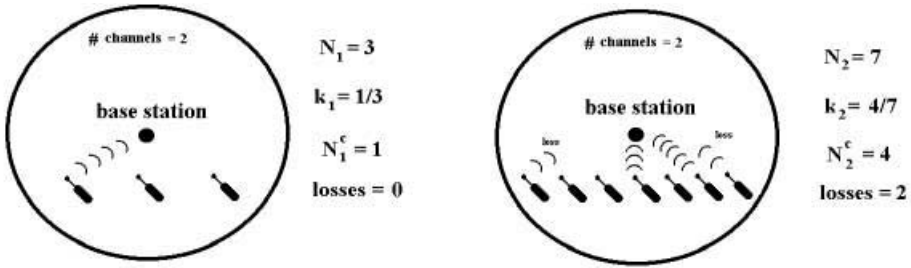


Fig. 1. The losses before and after traffic congestion.

We know that the movement of mobile users may be irregular, and thus prediction of their behavior is very difficult. However, we can observe and analyze the number of calls in a cell, which is considered as a random value. If the demand is steady we have a sequence of random values under one distribution. Otherwise, the distribution of the number of calls is changeable. Usually the following assumption is acceptable, where the number of calls ξ in a unit time is independent and identically distributed Poisson random value [7].

$$P(\xi = k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad k = 0, 1, \dots \quad (1)$$

The number of calls before traffic congestion has the Poisson distribution with parameter λ_1 , and it has the same distribution with rate λ_2 after traffic congestion occurs. It is necessary to detect the demand change, in the strict sense, the moment when the distribution change occurs should be found. The detection problem is known as a point-of-change problem or a discord problem. It is a very popular problem [8, 9], and minimization of the time between the moment of discord and the moment of alarm is an important issue. As far as the channel resource is available, the quality of service is satisfied and the delay for the discord detection can be tolerated. In this paper, we focus on a point-

of-change problem with acceptable delay, and propose a technique based on a point-of-change problem with admissible lag.

3 The Point-of-Change Problem with Poisson Distribution

3.1 Problem Statement and Discord Detection Algorithm

Let a sequence of independent random values ξ_i , $i = 1, 2, \dots$, be under consideration. ξ_i denotes the number of calls in time unit i . Before discord, the distribution of ξ_i is modeled by the Poisson distribution of Eq. (1) with parameter λ_1 . In unknown point of time t_D the intensity of flow of calls is changed. It means that the distribution parameter of ξ_i becomes different from λ_1 . We denote the distribution parameter after the discord as λ_2 . Without loss of generality, we assume that $\lambda_1 \leq \lambda_2$. The other case can be easily modeled with simple modification.

It is desirable and sometimes required to alarm system managers if a discord takes place in advanced systems. Let us designate the moment of alarm as t_A and T be an admissible lag. Then we know that a discord is detected if $t_D < t_A \leq t_D + T$. If $t_A < t_D$, it must be a false alarm.

Now we propose the following algorithm for discord detection.

Discord Detection Algorithm

Step 1. Calculation of the sum $S_T = \sum_{i=1}^T \xi_i$.

Let us denote the variate S_T has Poisson distribution [10].

Step 2. If $S_n > h$, then the system raises an alarm and go to Step 5. Otherwise, proceed to the next step. Here n takes the value $T, T+1, \dots$. Note that h is a threshold value calculated prior to the beginning of the algorithm. It is considered in detail below.

Step 3. The following sum is computed

$$S_n = S_{n-1} + \xi_n - \xi_{n-T}, \text{ where } n = T+1, T+2, \dots$$

It is obvious that the variate S_n also has the Poisson distribution because S_n is a sum of independent Poisson random values. Hence, if the distribution parameter of ξ_i is λ , the distribution parameter of S_n becomes $T\lambda$.

Step 4. Go to Step 2.

Step 5. End.

The value of h decides the tradeoff between the rate of discord detection and false alarm. For example, if $h = 0$, discord will be unambiguously detected, but a false alarm will be generated on each observation. If h value is high, false alarm will be seldom declared, but the probability of discord detection will be very low.

Let us discuss the basic concept of the proposed method. The random values ξ_{n-T}, \dots, ξ_n are under consideration. Assume that the probability of a false alarm is fixed if the following restriction is true.

$$P(S_n > h, n < t_D) \leq \alpha. \quad (2)$$

Here α is a constant representing the degree of false alarm. Before applying the method, it needs to be decided according to how often false alarm can be tolerated.

The second issue is the number of discord omissions. A discord is lost if $S_n < h$ and $n \geq t_D + T$. Hence, the probability of omission is

$$P(S_n < h, n = t_D + T). \quad (3)$$

For the quality of discord detection algorithm, it needs to minimize the probability of Eq. (3), while taking into account Eq. (2). It is achieved with an optimal choice of threshold value h_{opt} .

Theorem 1. *For the proposed method the optimal choice of threshold is*

$$h_{opt} = \min\{h : \sum_{k=0}^h \frac{(\lambda_1 T)^k}{k!} > e^{\lambda_1 T} (1 - \alpha)\}. \quad (4)$$

The probability of omission is then

$$\sum_{k=0}^{h_{opt}} \frac{(\lambda_2 T)^k}{k!} e^{-\lambda_2 T}.$$

Proof: Let us consider Step 2 of the algorithm. The decision rule is true, but the discord is not found ($n < t_D$). Hence, a false alarm takes place and S_n has the Poisson distribution with parameter $T\lambda_1$. It is clear that S_n is a positive integer. Hence, it can be assumed that h is a positive integer. We have

$$P(S_n > h, n < t_D) = 1 - P(S_n < h, n < t_D) = 1 - \sum_{k=0}^h \frac{(\lambda_1 T)^k}{k!} e^{-\lambda_1 T} \leq \alpha.$$

Thus, if the condition of Eq. (2) is true, then the threshold value belongs to a set

$$\{h : \sum_{k=0}^h \frac{(\lambda_1 T)^k}{k!} \geq (1 - \alpha) e^{\lambda_1 T}\}. \quad (5)$$

Without loss of generality we assume that $t_D = 0$. An omission of point-of-change occurs if $S_n < h$ and $n \geq T$. The aim of the proposed method is to minimize the probability of Eq. (3) under the condition of Eq. (2). Hence, it needs to minimize the probability $P(S_T < h)$. h can take a value from the set of Eq. (5). Obviously, the discord omission probability decreases if the threshold decreases, but the probability of false alarm increases. So, an optimal value of threshold is the least possible value of the set of Eq. (5). Hence, we have Eq. (4).

At the same time

$$P(S_T \leq h_{opt}) = \sum_{k=0}^{h_{opt}} P(S_T = k) = \sum_{k=0}^{h_{opt}} \frac{(\lambda_2 T)^k}{k!} e^{-\lambda_2 T}.$$

This contradiction proves the theorem. \square

The following theorem is needed for the sequel.

Theorem 2. *h_{opt} does not depend on λ_2 . That is, the distribution after discord does not affect the optimal threshold value.*

The proof directly follows Eq. (4).

3.2 Varying Parameter after Discord

The rate of distribution after discord is a constant λ_2 as defined earlier. Sometimes it is possible to have unsteady rate such that the random value ξ_{t_D+1} has a Poisson distribution with the rate $\lambda_2 - \delta_1$, ξ_{t_D+2} with the rate $\lambda_2 - \delta_2, \dots, \xi_T$ with the rate $\lambda_2 - \delta_T$. Here, we have $\delta_i < \lambda_2$. It is possible that $\delta_j = 0, \delta_{j+1} = 0, \dots, \delta_T = 0; 0 \leq j \leq T$. In this case the quality of the proposed method is described by the following theorem.

Theorem 3. *The probability of discord omission is*

$$P(S_T < h_{opt}, t_D = 0) = \sum_{k=0}^{h_{opt}} \frac{(\lambda_2 T - \Delta)^k}{k!} e^{-(\lambda_2 T - \Delta)}, \Delta = \sum_{i=1}^T \delta_i.$$

Here h_{opt} is calculated using Eq. (3).

Proof: By Theorem 2, h_{opt} is obtained by Eq. (3). Since ξ_1, \dots, ξ_T are independent Poisson random values, the sum of those values, S_T , has the Poisson distribution too. The parameter of the distribution of S_T is equal to $T\lambda_2 - \Delta$. So,

$$P(S_T < h_{opt}, t_D = 0) = \sum_{k=0}^{h_{opt}} \frac{(\lambda_2 T - \Delta)^k}{k!} e^{-(\lambda_2 T - \Delta)}. \quad \square$$

4 Performance Evaluation

In this section we analyze the quality of the proposed method through computer simulation. For it, a pseudorandom generator for Poisson distribution is necessary, and the choice of best generator is discussed. Also, we compare the analytical data and simulation data, and verify that the degree of false alarm is limited. The proposed algorithm is tested for both the constant and varying parameters of the distribution. For each simulation experiment, 10^6 runs are averaged to show the sensitiveness of the algorithm to the lag and parameters of distributions.

4.1 Pseudorandom Generator for Poisson Distribution

For the performance evaluation of mobility management such as mobile location management and hand off schemes, simulation is frequently employed. Usually the ranges of parameter values of wireless network are quite large, and it results in complex calculation and large simulation time. Therefore, it is important to choose a fast pseudorandom number generator.

In this paper we use a sequential random values of Poisson distribution for modeling the number of calls per unit time. There exist many algorithms generating Poisson pseudorandom numbers [10–12]. They are compared taking into account the specific property of the target problem.

The methods in [10, 11] use the idea that if the time between arrivals is exponentially distributed, then the number of arrivals have the Poisson distribution. We also need to use a uniform pseudorandom number generator. The method [12] uses only one uniformly distributed numbers. In Table I below the result of the generators tested is given. It shows that the algorithm from [12] is the best for average and large values of the Poisson distribution parameter. Method (A) of Molloy and Knuth[10, 11] have an advantage for small value parameters.

Table I. The time for generating 1,000,000 pseudorandom numbers (nanosecond).

Distribution rate	Ermakov	Knuth; Molloy (A)	Knuth; Molloy (B)
0.0001	38	33	44
0.01	38	33	44
0.1	39	38	44
0.2	39	44	44
0.3	38	50	44
0.5	43	47	50
1	47	61	58
2	49	99	71
3	66	137	82
5	82	198	109
10	143	363	176
100	1027	3279	1373
300	3010	9694	4010
500	5141	16323	6712

4.2 Verification of Limited Degree of False Alarm

According to Eq. (2), the proposed method gives a small number of false alarm. In each test the method employs a random value T , and a counter is increased if a false alarm takes place. Let the degree of false alarm be equal to or less than 5% throughout the tests. An admissible lag is selected among different values.

Table II. The degree of false alarm.

λ_2	T	h_{opt}	False alarm(%)
0.01	20	5	2.0
1	10	15	5.0
2	10	28	3.4
3	10	39	4.8
3	20	73	4.3
3	50	170	4.9
3	5	22	3.3
3	3	14	4.0
3	1	6	3.5
10	10	117	4.2
50	10	537	4.8

Each row in Table II is a result of simulation experiment for different λ_2 values. At the beginning of each experiment, the optimal threshold is calculated based on **Theorem 1** and the value is used for all the runs. The number of false alarms is in the column for **false alarm** which is given in percentage. We see that the degree of false alarm is always lower than 5%.

4.3 The Efficiency of the Proposed Algorithm

Let us find out the quality of the proposed discord detection algorithm. Without loss of generality we assume that $t_D = 0$ for discord detection. So, we have a distribution with a discord at the beginning. The simulation method is the same as above and the degree of false alarm is equal to 5%.

The discord detection rates from the simulation and analytical model are shown in Table III. Notice that they are very close, which verifies the effectiveness of the proposed model. The proposed discord detection algorithm always detects a demand change for relatively large distribution parameter values even though λ_1 and λ_2 are close and the admissible lag is not large.

Let us now consider the case of non-constant rate after the point-of-change. Let a random value ξ_i , $i = 1, \dots, T$, after the discord have the parameter of distribution $\lambda_1 + \delta i$, if $\lambda_1 + \delta i < \lambda_2$. Otherwise, the rate is equal to λ_2 . In other words, the rate of distribution after the discord changes gradually with a constant step δ up to λ_2 . The results for $\delta = 0.1$ are given in Table IV.

Table III. The efficiency of the algorithm.

λ_1	λ_2	T	h_{opt}	Simulation	Model
1	1.2	10	15	15.7	15.6
1	1.2	100	117	58.2	58.5
1	1.2	300	329	94.9	94.8
1	3	10	15	100	100
2	3	20	28	87.1	86.5
3	5	15	56	98.8	98.7
2	4	30	73	100	100
5	10	20	117	100	100
10	11	100	346	100	100
100	120	10	346	100	100

Table IV. Non-constant rate after discord.

λ_1	λ_2	T	Simulation	Model
1	2	10	48.5	48.3
1	2	20	91.1	88.3
10	11	10	12.1	12.2
10	11	20	29.7	26.8
1	3	10	88.4	88.9
100	120	10	100	100

We see that the quality of the proposed method in the case of varying distribution parameter is worse than that of the constant case. However, it is not true if T is relatively large. In Figure 2 the relationship between the quality and T is demonstrated for the case of $\lambda_1 = 2$, $\lambda_2 = 3$, and $\delta = 0.1$.

5 Conclusion

It is important to know the number of calls for moving users for effective management of cellular networks. The number depends on the behavior of customers, which is usually irregular and very difficult to model. The method for detecting the demand change in terms of the numbers has been presented in this paper. The point-of-change problem under admissible lag has been stated and analyzed by computer simulation with Poisson random value. The proposed algorithm for detecting the demand change shows high accuracy under heavy load, and the developed model can accurately predict the performance of the algorithm.

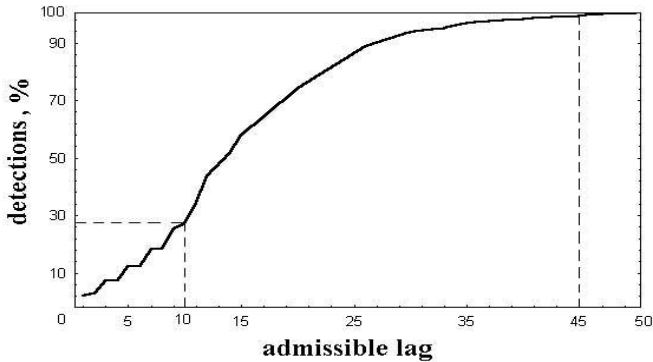


Fig. 2. The relationship between the detection rate(%) and admissible lag; $\lambda_1 = 2$, $\lambda_2 = 3$.

References

1. W. Lee, *Mobile Cellular Telecommunications: analog and digital systems*, Second Edition, New York: McGraw-Hill, 1995.
2. R. Thomas, H. Gilbert, and G. Mazziotto, "Influence of the Movement of Mobile Station on the Performance of the Radio Cellular Network," *Proceeding of the 3rd Nordic Seminar, Copenhagen*, September 1988.
3. C.Y. Park, Y.H. Han, C.S. Hwang, and Y.S. Jeong, "Simulation of a Mobility Prediction Scheme Based on Neuro-Fuzzy Theory in Mobile Computing," *Simulation*, v. 75, No 1, p. 6-17, 2000.
4. D. Hong and 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*, v. 35, No 3, pp. 77-91, August, 1986.
5. G. Liu and Jr. Maguire, "Class of Mobile Motion Prediction Algorithms for Wireless Mobile Computing and Communications," *Mobile Networks and Application*, v. 1, pp.113-121, 1996.
6. T. Lui, P. Bahl, and I. Chlamtac, "An Optimal Self-Learning Estimation for Predicting Inter-cell User Trajectory in Wireless Radio Networks," *IEEE 6th International Conference on Universal Personal Communications*, v. 2, pp.438-442, 1997.
7. Y. Fang, I. Chlamtac, and Y.-B. Lin, "Call completion probability for a PCS network", *IEEE 6th International Conference on Universal Personal Communications*, v. 2, pp. 567-571, 1997.
8. E.S. Page, *Continuous inspection schemes*. Biometrika, 1954, v.41, N2, p.100-114.
9. B. Yakir, "Dynamic sampling policy for detecting a change in distribution, with a probability bound on false alarm," *Ann. Statist.*, v. 24, pp. 2199-2214, 1996.
10. M. Molloy, *Fundamentals of Performance Modeling*, New York : Macmillan Publishing Company, 1988.
11. D.E. Knuth, *The art of computer programming*, 3rd ed., v.2, pp. 137, Addison-Wesley, 1998.
12. S.M. Ermakov and G.A. Michaelov, *Statistics modelling*, oscow: Nauka, 1982 (in Russian).