Combined Preemption and Queuing Schemes for Admission Control of a Wireless Emergency Network

Jiazhen Zhou and Cory Beard
Department of Computer Science/Electrical Engineering
University of Missouri-Kansas City
Contact: beardc@umkc.edu

Abstract— **Preemption is a possible way to ensure emergency traffic's priority in wireless network. But a pure preemption policy can be especially harsh to low priority calls. A way to improve this is introducing a queuing mechanism, which means putting preempted calls into a queue so that they can resume when channels become available, as long as they do not give up due to waiting too long in the queue. Also, we bring out a single preemption scheme, to avoid calls being preempted multiple times. The analytical results for performance analysis of these methods are given, and based on them the performance of different schemes are evaluated and compared.**

I. INTRODUCTION

Network congestion can happen due to a lot of reasons. In this paper, we mainly focus on disaster events as they cause congestion in wireless networks. After disaster events happen, tremendous stress is placed on networks due to the rise in traffic demand, including demand from public and emergency staff. As pointed out in [1,2,3], network demand can be up to 10 times of normal. Among the traffic demands, emergency traffic should be given special priority for saving life and property.

As studied in the admission control field, the priority can be implemented in different ways, and thus have different strengths. The strongest approach is preemption, which gives high priority calls immediate access, unless all channels are already taken by high priority ones. Also this makes admission of high priority calls virtually unaffected by an increase in low priority traffic demand. Early work for emergency traffic is seen in [5].

A weaker approach is a queueing method, where emergency calls (and only emergency calls) are put into a waiting queue, and get access to channels once they are free. Work similar to this is seen in [7], there they set the handoff traffic as the high priority ones.

The even weaker approach is to put both emergency calls and public calls into separate queues, and schedule the queues according to different scheduling schemes. An example for this is in [4] where both emergency traffic and public handoff traffic are put into queues, and scheduled according to a weighted earliest deadline scheme. Another example from real world application is the Public Use Reservation with Queuing All Calls policy (PURQ-AC) [2]. Here both emergency calls and public calls are queued, and upon a call's departure, the queues are scheduled in a way similar to a round-robin algorithm. This ensures that when congestion happens, a recommended 1/4 schedule is used so that emergency users can only take about 25% of the channels while the other 75% of the channels are taken by public users. The purpose of this policy is to avoid starving public users, but it does not differentiate handoff calls from originating calls.

Our work in this paper is to make the preemption policy not so "preemptive". The pure preemption approach is currently not allowed in today's networks in the United States [2], but in this paper we propose ways to use preemption while not being as harsh for public users since their calls can resume after a short time.

So we study possible strategies that can help low priority calls to get a better chance to be admitted. The basic idea is combining queuing with preemption, and we make following assumptions:(1) Same as mentioned in [4], the main three types of voice calls we are to deal with are emergency calls, public handoff calls and public originating calls, the latter two types of calls will also be mentioned as low priority calls in later sections. (2) Our study focuses on a single cell. (3) All call durations are independently, identically, and exponentially distributed. (4) There is no handoff for emergency calls; we assume most emergency users will be stationary within a disaster area. However, for assumption (4), the model given here can be easily extended to a more general situation.

A related work is in [8]; they studied two types of traffic: real-time (voice) and non-real-time (data). Each type consists of both originating and handoff traffic, and both handoff traffics have their own queues. They designed the network such that real-time handoff traffic can preempt resources from ongoing non-real-time traffic, and put the interrupted traffic into the non-real-time handoff traffic queue.

In our paper, all traffics we deal with are voice, and thus sensitive to waiting long in the queue whether they are queued handoff calls or preempted calls. Based on this fact we make detailed analysis of the expiration of calls in a queue (users become impatient and give up) and their effect on system performance. However, the behavior of expiration of calls in the queues was not studied in [8].

This work is supported by the United States National Science Foundation under CAREER Award ANI-0133605.

The main contributions of this paper include:

1) Introducing combined preemption and queuing methods for a network that supports emergency traffic. Performance metrics like blocking probability, preemption probability, and average numbers of preemptions per call are analyzed.

2) A priority modification policy is brought out to avoid multiple preemption. In essence, a low priority call becomes a high priority call once it returns from a queue, so it cannot be preempted again. Analytical results about performance metrics are given.

3) The behavior of combined preemption and queuing schemes, the pure queuing scheme, and the pure preemption scheme are compared, thus deep insight into the benefits and shortcomings of each policy are provided. Operators can decide which scheme is closer to their requirements.

This paper shows that preemption is a good way to ensure the emergency traffic's priority in a wireless cellular network, but the pure preemption policy seems to be too strong and too unfair for public traffic. Preemption with queuing can increase the chance of public traffic to succeed and makes the system resource more wisely used while not affecting the performance of emergency traffic. A policy that restricts preemptions to only happen once per call is even nicer to public traffic, but also affects emergency performance somewhat. We also show that preemption with queueing is better than a pure queuing policy.

In section II, different schemes and performance analysis is given, Section III compares these possible schemes, and in Section IV we conclude this paper.

II. Combined Preemption and Queuing Policies and Performance Analysis

A. Call Preemption and Queuing with 2 Queues

In Fig. 1 we illustrate the combined preemption and queuing scheme. Class 1 is made of emergency calls, Class 2 is handoff calls, and Class 3 is calls originating from within cell that are low priority. There are separate queues for handoff and preempted calls. To facilitate analysis, we assume that the service time of preempted calls are renewed, and the expiration time is always the same after each resumption. (In the real life the user can become more and more impatient after multiple times of preemption. And more impatient the user is, larger chance will the user drop the call. This will be studied in later work)

With two queues there, we can have different choices to schedule calls in the queue when channels are available. In this paper, we use priority queueing and assume that handoff calls have higher priority. In later work, we can use the weighted earliest deadline scheduling scheme [4] which provides flexibility for operators.

The state diagram for this scheme is a 3-dimensional Markov chain. To make it clear, we just give the example with 2 channels, and the length of both queues are 1. This is shown in Fig. 2. Each state is identified as (i,j,k,l), while i means channels taken by emergency calls, and j means the channels taken by low priority ones, k is the number of handoff calls in queue 1, and l is the number of preempted calls in queue 2.

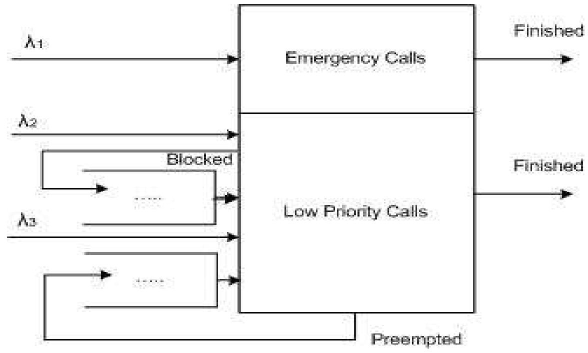

Fig. 1. Preemption and Queuing using separate queue for preempted calls

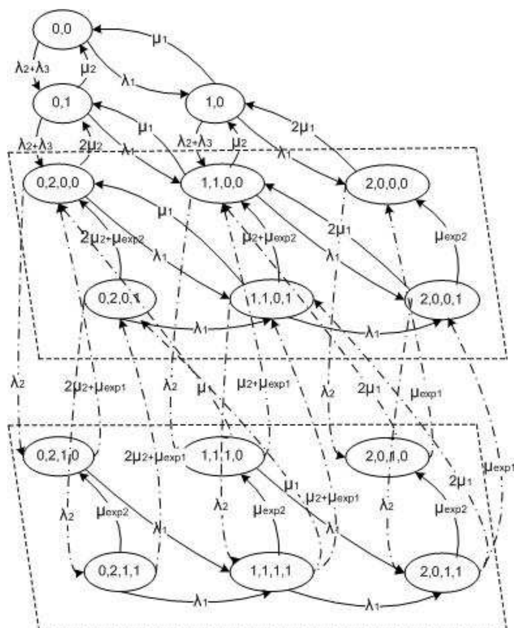

Fig. 2. State diagram for using separate queue for preempted and handoff calls

The main performance metrics we concern include: admission probability and success probability (i.e., probability of finishing normally without expiring or dropping) for each class, preemption probability for a low priority call given that it is admitted, and the average number of times that a call is preempted. To provide the analysis for these performance metrics, we also find related ones like blocking probability of each class, and the expiration probability of calls in the queue.

(1) Blocking probability

When an emergency call arrives and finds no channel immediately available or no channels to preempt (all ongoing calls are emergency ones), it will be blocked directly. For a handoff call, we say it will be blocked only when the queue is full, while for an originating call, it will be blocked whenever the channels are all occupied. From Fig. 2 we can see that:

$$P_B^1 = \sum_{k=0}^{L_1} \sum_{l=0}^{L_2} P(C, 0, k, l) \quad (1)$$

$$P_B^2 = \sum_{i=0}^{C} \sum_{l=0}^{L_2} P(i, C-i, L1, l) \quad (2)$$

$$P_B^3 = \sum_{i=0}^{C} \sum_{k=0}^{L_1} \sum_{l=0}^{L_2} P(i, C-i, k, l) \quad (3)$$

(2) Expiration probability

Calls in the queue will expire if not served or resumed in time. Expiration probability is defined as the total expired calls divided by the total arrived calls to the queue; alternatively the total numbers in the numerator and denominator can be replaced by rates. The arrivals to the queues include preempted calls and handoff calls. The preempted calls are caused by the arrival of emergency call when all channels busy and some are taken by low priority ones, so the rate is $\lambda_1 \sum_{k=0}^{L-1} \sum_{i=0}^{C-1} P(i, C-i, k)$. The handoff calls will enter into their queue when all channels are busy and the handoff queue is not full, so the rate of handoff calls into their queue is $\lambda 2(P_B^3 - P_B^2)$. The expiration probability for handoff queue and preempted queue is calculated as following:

$$P_{Exp}^{Ho} = \frac{Expired\ calls\ in\ queue\ 1/sec}{Arrival\ calls\ to\ queue\ 1/sec} = \frac{\sum_{i=0}^{C} \sum_{k=1}^{L_1} \sum_{l=0}^{L_2} P(i, C-i, k, l) k \mu_{exp}^1}{\lambda 2(P_B^3 - P_B^2)} \quad (4)$$

$$P_{Exp}^{Prm} = \frac{Expired\ calls\ in\ queue\ 2/sec}{Arrival\ calls\ to\ queue\ 2/sec} = \frac{\sum_{i=0}^{C} \sum_{k=0}^{L_1} \sum_{l=1}^{L_2} P(i, C-i, k, l) l \mu_{exp}^2}{\lambda 1(P_B^3 - P_B^1 - P_{Drp}^{Prm})} \quad (5)$$

(3)Preemption probability and average preemption times

The overall preemption probability for ongoing low priority calls is equal to the rate of calls preempted divided by the rate of low priority calls activated. The activated calls consist of two parts: the calls directly accepted, and those activated from the queues (those calls which entered into the queues and did not expire). The activated calls consist of two parts: the calls directly accepted and those taken out from queues. The calls activated from queues include those which come out from the handoff queue and from the preempted queue, which can be expressed as: $Rate_{OutofQ} = \lambda 1(P_B^3 - P_B^1 - P_{PrmDrp})(1 - P_{Exp}^{Prm}) + \lambda 2(P_B^3 - P_B^2)(1 - P_{Exp}^{Ho})$. P_{PrmDrp} is the probability that a call is preempted and dropped, which is defined as: $P_{PrmDrp} = \sum_{i=0}^{C-1} \sum_{k=0}^{L_1} P(i, C-i, k, L_2)$, while the probability that a call is dropped given that it's preempted is:

$$P_{Drp}^{Prm} = \frac{P_{PrmDrp}}{P_{Prm}} \quad (6)$$

Now we can calculate the preemption probability as follows:

$$P_{Prm} = \frac{Preempted\ calls/sec}{Activated\ calls/sec} = \frac{\lambda 1(P_B^3 - P_B^1)}{(\lambda 2 + \lambda 3)(1 - P_B^3) + Rate_{OutofQ}} \quad (7)$$

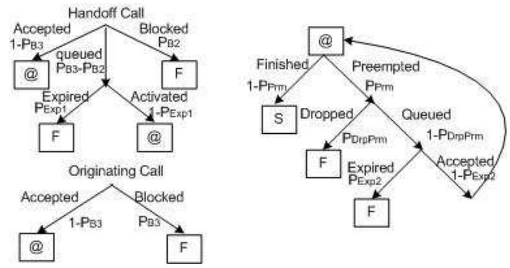

Fig. 3. Probability flow for multiple preemptions case

The average preemption times (number of times a call is preempted) given that a call is accepted can be computed in two ways. One is according to the definition:

$$\overline{PrmTimes} = \frac{Preempted\ calls/sec}{Accepted\ calls/sec} = \frac{\lambda 1(P_B^3 - P_B^1)}{(\lambda 2 + \lambda 3)(1 - P_B^3) + \lambda 2(P_B^3 - P_B^2)(1 - P_{Exp})} \quad (8)$$

Here the accepted calls means calls accepted to be serviced, each call will only be counted once even if it is preempted and resumed multiple times. Another way is based on the preemption and expiration probability we already derived above. In Fig. 3 we show the probability flow of low priority calls. In the frame, F means failed, S means successful, while "@" symbols in the two left trees represent a jump to the "@" symbol in the right tree. The right tree represents what might happen after a call is accepted. We notice that a call can be preempted multiple times, and the sub-flow is recursive here. From the figure we can get the probability of being preempted for 1 time, 2 times, ... n times, which is a geometric process with $Pr(Preempted\ n\ times) = P_{Prm}(1-A)A^n$, while $A = (1 - P_{Drp}^{Prm})(1 - P_{Exp})P_{Prm}$, then we know the expected value is $\frac{P_{Prm}}{1-A}$. So we get:

$$\overline{PrmTimes} = \frac{P_{Prm}}{1 - P_{Prm}(1 - P_{Exp})P_{Drp}^{Prm}} \quad (9)$$

Since the average preemption times we calculated above can be less than 1 or greater than 1, it is difficult to compare different scenarios or schemes. Here we use another concept– *relative average times*, which is defined as the average preemption times given a call is preempted:

$$Relative\ Average\ Preemption\ Times = Average\ Preemption\ Times | Call\ preempted = \frac{1}{1 - P_{Prm}(1 - P_{Exp})P_{Drp}^{Prm}} \quad (10)$$

(4) Average waiting time

To compute the average waiting time in the queues, Little's law will be applied after getting the average number in the queue and the average arrival rate. From Fig. 2 we can get: the average queue lengths are computed as: $\overline{L_{Ho}} = \sum_{k=1}^{L_1} \sum_{i=0}^{C} \sum_{l=0}^{L_2} P(i, C-i, k, l)k$, $\overline{L_{Prm}} = \sum_{l=1}^{L_2} \sum_{i=0}^{C} \sum_{k=0}^{L_1} P(i, C-i, k, l)l$. The average arrival rate to handoff and preempted queues is: $\lambda 1(P_B^3 - P_B^1 - P_{PrmDrp})$

and $\lambda 2(P_B^3 - P_B^2)$ individually. So:

$$\overline{T_{Ho}} = \frac{\sum_{k=1}^{L_1}\sum_{i=0}^{C}\sum_{l=0}^{L_2} P(i, C-i, k, l)k}{\lambda 1(P_B^3 - P_B^1 - P_{PrmDrp})} \quad (11)$$

$$\overline{T_{Prm}} = \frac{\sum_{l=1}^{L_2}\sum_{i=0}^{C}\sum_{k=0}^{L_1} P(i, C-i, k, l)l}{\lambda 2(P_B^3 - P_B^2)} \quad (12)$$

(5)Admission Probability

Now we show how to calculate the probability of calls to be admitted for each class. For an emergency call, it will be admitted if not blocked. Originating calls will be admitted if not directly rejected, while handoff calls can be admitted in two ways: directly admitted or admitted after waiting some time in the queue. The directly admitted part is equal to the admission probability of originating calls, the admission after waiting part is $(P_B^3 - P_B^2)(1 - P_{Exp}^{Ho})$, which means those that entered into the queue and not expire. So,

$$P_{Adm}^{Emr} = 1 - P_B^1 \quad (13)$$
$$P_{Adm}^{Orig} = 1 - P_B^3 \quad (14)$$
$$P_{Adm}^{Ho} = 1 - P_B^3 + (P_B^3 - P_B^2)(1 - P_{Exp}^{Ho}) \quad (15)$$

(6) Success probability

For emergency calls, all of the admitted calls will be successfully finished, thus *satisfying the dependability requirement*. But for low priority calls, this kind of dependability can not be assured. To compute the successfully finished probability with preemption possible, from Fig. 3 we can see that firstly we need to get $Pr(A\ call\ will\ succeed|Accepted)$. This can be computed as:

$$\begin{aligned} &Pr(Succ|Accpt) \\ &= (1 - P_{Prm})\sum_{i=0}^{\infty}(P_{Prm}(1 - P_{Drp}^{Prm})(1 - P_{Exp}))^i \\ &= \frac{(1 - P_{Prm})}{1 - P_{Prm}(1 - P_{Drp}^{Prm})(1 - P_{Exp}^{Prm})} \end{aligned} \quad (16)$$

Then obviously:

$$\begin{aligned} P_{Succ}^{Orig} &= Pr(Succ|Accpt)Pr(Orig.\ Call\ Accpt) \\ &= Pr(Succ|Accpt)(1 - P_B^3) \end{aligned} \quad (17)$$

For a handoff call, it can be accepted directly or accepted after queueing for some time in the queue. So:

$$\begin{aligned} &P_{Succ}^{Ho} = Pr(Succ|Accpt)Pr(Handoff\ Call\ Accpt) \\ &= Pr(Succ|Accpt)((1 - P_B^3) + (P_B^3 - P_B^2)(1 - P_{Exp}^{Ho})) \end{aligned} \quad (18)$$

B. Single Preemption Scheme

For the combined preemption and queuing policy given above, we can see that once a call is preempted, nothing stops it from possibly being preempted many times. This is annoying to users and may cause more calls to be abandoned than we have shown in the above analysis. To avoid it, we can increase the priority of calls after it's resumed from preemption queue. By introducing a priority change mechanism, we also hope to improve the low priority user's chance of finishing the call while not harming the total system's performance. Since using this policy there can be at most one time that a preemption happens, we call it as single preemption policy, correspondingly the basic combined preemption and queuing policy is called as multiple preemption scheme in later sections. The idea of this method is shown in Figure 4. The state diagram for the single preemption policy is basically the same as the diagram of the multiple preemption case; the only difference is that when there is a channel free and no calls are waiting in the handoff queue, the preempted calls will be resumed and become high priority calls which cannot be preempted again. The definition for state (i,j,k,l) is the same as the multiple preemption case.

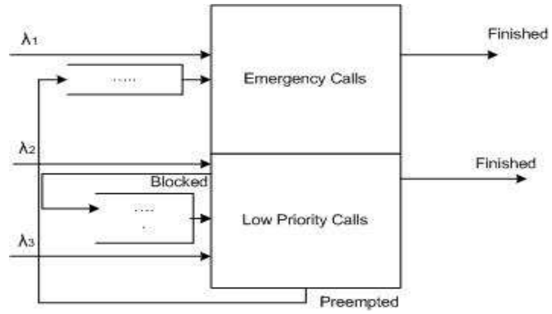

Fig. 4. Preemption and Queuing using separate queue for preempted calls

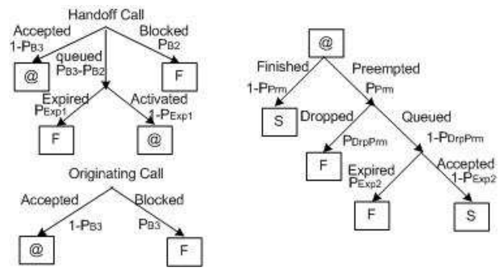

Fig. 5. Probability flow for single preemption case

Most computations here are similar to multiple preemption case. Here we just show those that are different:

(1)Preemption probability

The only difference from multiple preemption for computing preemption probability lies in the average rate out of queue. Because calls taken from the preempted queue become high priority ones and will never be preempted again, we have $Rate_{OutofQ} = \lambda 2(P_B^3 - P_B^2)(1 - P_{Exp}^{Prm})$

In addition, since a call can be preempted one time at most, the expected value of preemption times is the same as preemption probability, and the relative preemption times is always 1.

(2) Successfully finished probability According to Fig. 5 we can get the success probability easily. For an accepted call, if it finished directly or is preempted once and then resumes (becomes a high priority call, surely to be finished successfully), then it's regarded as being a success. We can get $Pr(Succ|Accpt) = (1 - P_{Prm}) + P_{Prm}(1 - P_{Drp}^{Prm})(1 -$

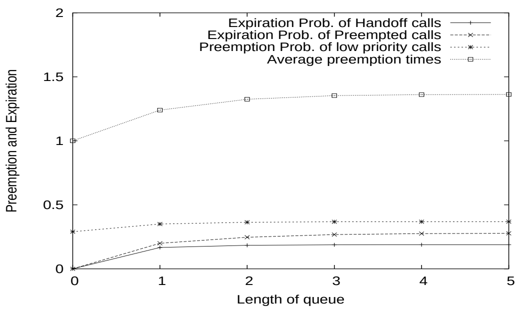

Fig. 6. Comparison of queuing vs. no queuing for preemption and expiration prob.

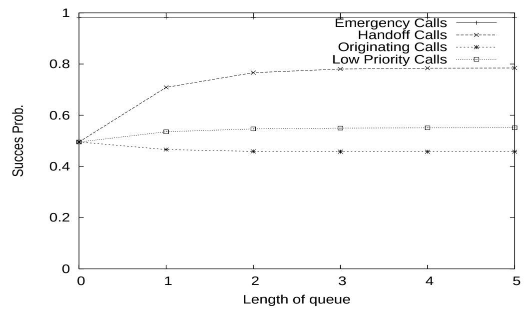

Fig. 7. Comparison of queuing vs. no queuing for success prob.

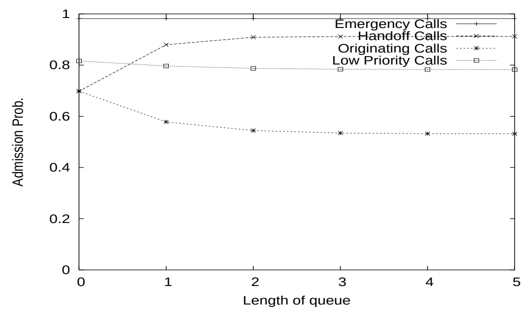

Fig. 8. Comparison of queuing vs. no queuing for admission prob.

P_{Exp}^{Prm}), So:

$$P_{Succ}^{Orig} = (1 - P_B^3)Pr(Succ|Accpt) \quad (19)$$

$$P_{Succ}^{Ho} = P_{Succ}^{Orig} + (P_B^3 - P_B^2)(1 - P_{Exp}^{Ho})(1 - P_B^3)) \quad (20)$$

III. Comparison Between Different Policies

A. System Utility Evaluation

To facilitate the comparison between different policies, we introduce two new concepts: Admitted system utility and Successful system utility, which are defined as

$$SysUtilA = \sum_{i=1}^{3} \frac{\lambda_i P_{Accpt}^i}{C\mu} \quad (21)$$

$$SysUtilS = \sum_{i=1}^{3} \frac{\lambda_i P_{Success}^i}{C\mu} \quad (22)$$

$SysUtilA$ means the system utility with regard to admitted calls, and $SysUtilS$ represents the system utility with regard to successfully finished calls. There are some calls admitted but terminated before normal ending because of failing to enter into the queue after preemption, or waiting too long in the queue so that they lose the patience and are abandoned (expired). Expired calls will occupy some system resources, and this can be viewed as kind of waste of resource because the termination will cause dissatisfaction.

It's obvious that $SysUtilA \geq SysUtilS$ and $SysUtilS \leq 1$. The closer $SysUtilS$ is to 1, or the closer is $SysUtilA$ to $SysUtilS$, the better is the chance for the channels to be taken by those calls which successfully finish after being admitted.

B. Combined policy vs. Pure preemption policy

With a queuing policy added to the preemption method, the system needs to add memory and will have more to manage. But what's the benefit? And will emergency call's performance be affected? In this subsection we will show the effect of the queuing policy with preemption. To facilitate comparison with pure queuing and pure preemption policy, the basic combined preemption and queuing policy is mentioned as *combined policy*.

In making comparisons, generally we make most parameters fixed and only one parameter will change. The basic parameters we take are: $\mu 1 = \mu 2 = 0.01(average\ service\ time\ is\ 100\ seconds)$, Channel number $C = 10$, arrival rates:$\lambda_1 = 0.05, \lambda_2 = 0.02, \lambda_3 = 0.05$, expiration rate $\mu_{exp}^1 = \mu_{exp}^2 = 0.02(50\ seconds)$, and queue length $L_1 = L_2 = 5$. The rates of originating and handoff traffic are assumed to be fixedly related with $\lambda_2 = 0.4\lambda_3$

Fig. 6, 7 and 8 show how the performance metrics change when the queuing policy is applied, we can see that:

(a) As shown in Fig. 7, the total success probability is improved when queuing policy is applied, and it keeps increasing when queue length become longer, yet the improvement is slower and slower so that after some point, an increase in queue length does not have much effect. Note that a pure preemption policy corresponds to the length of the queue equal to zero.

(b) With queue or no queue makes a big difference for handoff and originating calls: making handoff calls have better chance to be admitted and succeed while making originating calls have a worse chance.

(c) Fig. 8 show that the admission probability of low priority calls is lower when queuing policy is applied, while emergency calls' doesn't change. So we can conclude that the total admission probability goes down when the queueing policy is applied and when queue length increases.

(d) From Fig. 6, preemption probability for low priority calls is also higher when queuing policy is applied. This shows that with a queuing policy added, low priority calls might be preempted more often and need to wait some time in the queue, but also have a better chance to finish.

So, *the introduction of queues makes fewer low priority calls to be admitted, but more to succeed. This shows that the system resources are more wisely used, and also proves that the combined preemption and queuing policy works better*

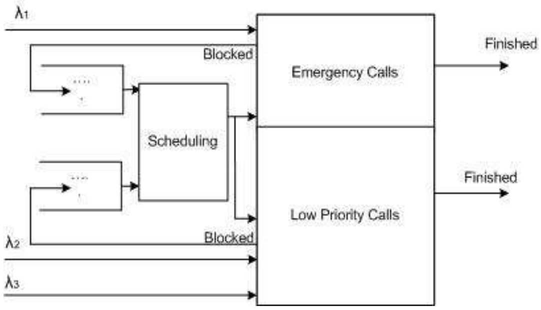

Fig. 9. Queuing and scheduling policy

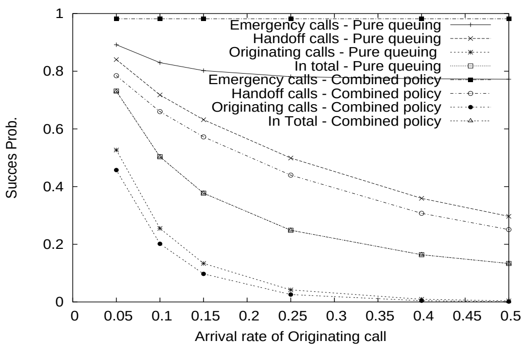

Fig. 10. Comparison of pure queueing and preemption with queueing: λ_3 vs. success prob.

than a preemption policy by itself.

C. Combined policy vs. Pure queuing policy

From the above subsection we can see that the combined preemption and queuing policy is indeed an improvement upon a pure preemption policy. Now we will see how it is compared with a pure queuing policy. A pure queueing policy is what is widely used today for emergency calls.

In our work for pure queuing policy [4], we use two queues, one for emergency and one for handoff. This is shown in Fig.9. The scheduling scheme we used is weighted earliest deadline scheduling, which can leave the choice to operators to decide either emergency or handoff calls have higher priority, and how much higher is the priority. This is accomplished by setting the weighting parameter.

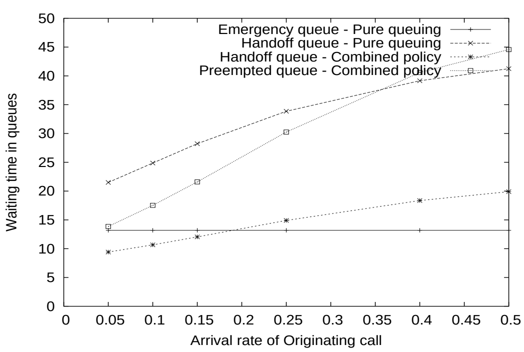

Fig. 11. Comparison of pure queueing and preemption with queueing: λ_3 vs. average waiting time

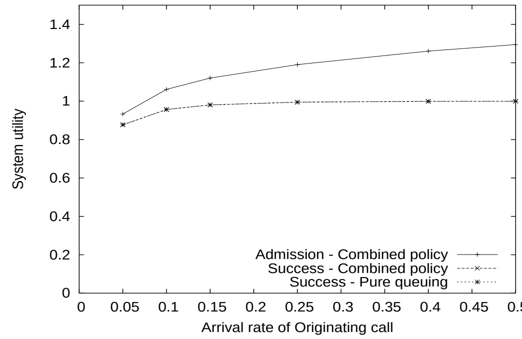

Fig. 12. Comparison of pure queueing and preemption with queueing: λ_3 vs. system utility

In this experiment for the pure queuing case, we take the weighting parameter as 0, which means it's a priority queue and emergency calls have the priority over handoff calls. In Fig. 10,11 and 12 we showed the performance change according to originating traffic for both policies (recall that handoff traffic is in fixed proportion to originating traffic). We can see that:

(a) From Fig. 10, when low priority traffic (combination of handoff and originating) goes up, the success probability of the preemption and queuing scheme doesn't change while the pure queuing policy case keeps dropping.

(b) The total success probability for both schemes is almost the same despite how low priority traffic changes. The curves lie virtually on top of each other.

(c) Using the system utility concept we defined above, through Fig. 12 we find that the success system utility for both schemes also stays almost the same. So both schemes provide the same efficiency in using system resources. The admission system utility for the preemption and queuing case is higher, and it can be higher than 1 when the system becomes very congested, which means that some calls are admitted but dropped before they succeed to finish (the system admits more than it can handle).

(d) As shown in Fig. 11, for a pure queuing policy, the average waiting time in the queue for emergency calls keeps steady when the low priority traffic changes. But for handoff calls the waiting time keeps increasing. For the combined preemption and queuing policy, there is no queue for emergency calls, so they also do not need to wait before getting admitted.

So, we can conclude that *compared with a pure preemption policy, the total system performance is almost the same. But the preemption and queuing policy makes the emergency calls' performance more guaranteed and waiting time equal to zero because the priority given to emergency calls is much stronger. As a tradeoff, both handoff and originating calls get degraded performance with preemption and queuing.*

D. Multiple preemption vs. Single preemption

Now, we will see the difference in the behavior between the multiple preemption and the single preemption policy. In Fig. 13, 14, 15, we show the success probability when λ_3, queue length, and expiration rate change. We can see that:

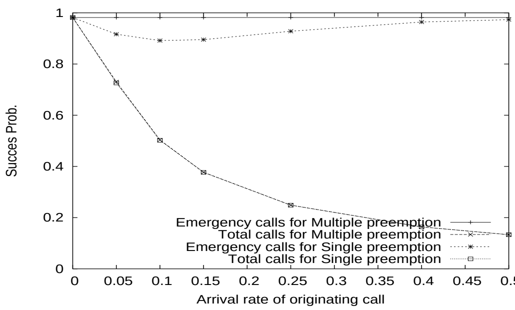

Fig. 13. Comparison of single and multiple preemption:Low priority traffic vs. success probability

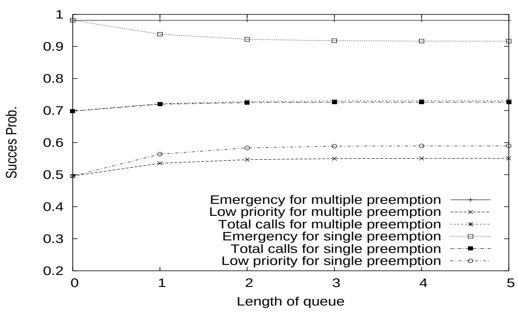

Fig. 14. Comparison of single and multiple preemption: Queue length vs. success probability

(1) When the low priority traffic goes up, the admission probability of emergency calls is not affected in the multiple preemption case. But in the single preemption case, the emergency calls are affected. We notice that the admission of emergency calls will go down at first because some resources are taken by low priority calls and can't be preempted. But when low priority traffic keeps going up, the emergency call's admission probabilities go up instead. This is because that after system is congested, more and more preempted calls are blocked after being preempted or expire in the preempted queue, so the calls resumed from the preempted queue are less and less and the chance for emergency calls to be admitted is better and better.

(2) We can also see that the admission probability of emergency calls in the single preemption case is always lower than the multiple preemption case. But the success probability of the total system is the same for both policies as low priority traffic changes.

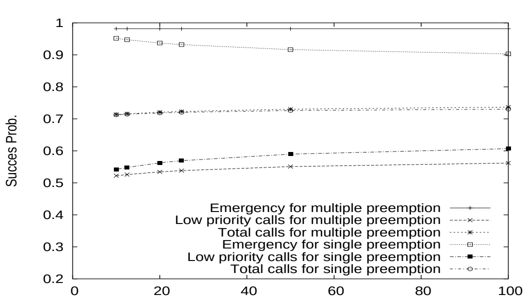

Fig. 15. Comparison of single and multiple preemption: Expiration time vs. success probability

(3) Admission of emergency calls in the single preemption case is also affected by the queue length and average expiration time, while in multiple preemption case, it's not.

So, we can see that *with single preemption policy applied, the success probability of low priority calls are improved upon multiple preemption case, while the admission probability of emergency calls is not so guaranteed as in multiple preemption case, but still good compared with low priority calls.*

IV. CONCLUSION

Preemption is a good way to ensure priority to emergency traffic in a wireless cellular network. Yet the pure preemption policy seems to be too strong and too unfair for public traffic, and thus is generally forbidden in real life networks. This paper showed that the introduction of a queuing policy can increase the chance of public traffic to succeed and make the system resources more wisely used while not affecting the performance of emergency traffic. Through comparisons, we also showed why this policy is better than a pure queuing policy.

Yet there is multiple preemption problem in the preemption and queuing policy. To remove the multiple preemption problem, we bring out a new policy which only allows the ongoing public call to be preempted at most once. Then we find that the emergency call's admission will be affected, and as the tradeoff, public users have much better chance to finish the call. This provides another possible scheme to operators because public users can benefit more from it. Preemption with queuing is a viable option to consider in today's cellular systems, whether allowing multiple preemptions or single preemptions.

REFERENCES

[1] Federal Emergency Management Agency, "Technology Applications by the Federal Emergency Management Agency in Response, Recovery, and Mitigation Operations." Paper Presented to the 27th Joint Meeting of the U.S./Japanese Panel on Wind and Seismic Effects, Tokyo/Osaka, Japan, May 16-27, 1995.

[2] Nyquetek Inc., "Wireless Priority Service for National Security / Emergency Preparedness: Algorithms for Public Use Reservation and Network Performance," August 30, 2002. Avaiable at http://wireless.fcc.gov/releases/da051650PublicUse.pdf.

[3] S. Adamson and S. Gordon, "Analysis of Two Trunk Congestion Relief Schemes," *Proceedings of IEEE MILCOM '93*, pp. 902-906, 1993.

[4] J. Zhou and C. Beard, "Weighted Earliest Deadline Scheduling and Its Analytical Solution for Admission Control in a Wireless Emergency Network." *International Teletraffic Conference(ITC19)*, August 29-September 2, 2005, Beijing.

[5] C. Beard, "Preemptive and Delay-Based Mechanisms to Provide Preference to Emergency Traffic." *Computer Networks Journal*, Vol. 47:6, pp. 801-824, 22 April 2005.

[6] C. Beard and V. Frost, "Prioritized Resource Allocation for Stressed Networks," *IEEE/ACM Transactions on Networking*, Vol. 6, no. 5,October 2001, pp. 618-633.

[7] D. Hong and S. S. Rappaport, "Traffic Model and Performance Analysis for Cellular Mobile Radio Tele- phone Systems with Prioritized and Non-prioritized Handoff Procediires," *IEEE Trans. on Vehicular Technology*, vol. VT-35, no. 3, pp. 77-92, Aug. 1986.

[8] J. Wang, Q.-A. Zeng, and D. P. Agrawal "Performance Analysis of Preemptive Handoff Scheme for Integrated Wireless Mobile Networks," *Proceedings of IEEE GLOBECOM'01, San Antonio, TX, 2001, pp. 3277 - 3281.*