Achievable region with impatient customers

Veeraruna Kavitha
IEOR, Indian Institute of Technology Bombay, India
vkavitha@iitb.ac.in

Raman Kumar Sinha
IEOR, Indian Institute of Technology Bombay, India
ramankumar2112@gmail.com

ABSTRACT

We consider a queueing system with heterogeneous agents. One class of agents demand immediate service, would leave the system if not provided. The second class of customers have longer job requirements and can wait for their turn. We discuss the achievable region of such a two-class system, which is the region of all possible pairs of performance metrics. Blocking probability is the relevant performance for eager/impatient class while the expected sojourn time is appropriate for the tolerant class. We obtain the achievable region, considering static policies that do not depend upon the state of the second class, using a conjecture of a pseudo conservation law. This law relates the blocking probability of eager customers with the expected sojourn time of the tolerant customers, in a fluid limit for eager customers. We validate the pseudo conservation law using two example families of static schedulers, by deriving their performance. Along the way, we obtain smooth control (sharing) of resources between such heterogeneous classes (e.g., voice and data calls of a communication system). We also demonstrate that the dynamic achievable region (obtained using state dependent policies) is strictly bigger than the static region, by deriving the performance of an example dynamic policy.

Index terms– Heterogeneous users, achievable region, processor sharing, dynamic and static schedulers.

CCS CONCEPTS

• **Mathematics of computing**;

ACM Reference Format:
Veeraruna Kavitha and Raman Kumar Sinha . 2017. Achievable region with impatient customers. In *VALUETOOLS 2017: 11th EAI International Conference on Performance Evaluation Methodologies and Tools, December 5–7, 2017, Venice, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3150928.3150953

1 INTRODUCTION

We consider queueing systems with heterogeneous classes of users. One is a class of eager/impatient customers, who would quit the system if service is not offered immediately. The other class of customers are tolerant, can wait for their turn. An achievable region for an n-class system is the set of all possible relevant performance vectors $(pm_1, \cdots, pm_n)$, obtained by varying all possible scheduling policies ([3, 5] etc.,). Our aim is to study the achievable region for a queuing system with heterogeneous classes, under certain conditions. The important performance metric for impatient customers is the blocking probability, the probability that a customer returns without service. The tolerant customers can wait for their turn, however their satisfaction depends upon the expected sojourn time (the total time spent by a typical customer in the system). This paper focuses on heterogeneous achievable region, the set of all possible pairs of blocking probability and expected sojourn time.

One of the main motivations for this paper is data-voice calls of a communication network. Data calls are delay tolerant, but require precision. They can tolerate delays in the service, but not errors in transmission. Their job requirements are usually long. On the other hand the voice calls are impatient, need immediate service. However their job requirements would usually be smaller. We consider two policies for capacity/resource sharing between the data-voice calls. In the first policy entire capacity is transferred to the voice calls (when admitted), irrespective of the number of users receiving the service. The voice calls operate in processor sharing mode, and we refer to this as *PS* policy. In the second (capacity division/*CD*) policy, the capacity transferred (resources allocated) to voice calls is proportional to the number of users receiving the service.

Consider a communication system with K orthogonal channels. For example, each channel could be one or a collection of resource blocks as in an OFDM based LTE network (e.g.,[1]). Initially all the channels are dedicated to data calls. As and when the voice calls arrive, one by one the channels are transferred and the data calls use the remaining. Our *CD* policy captures this scenario precisely. If the voice calls are served at the highest possible rate as in *PS* policy, it improves the chances of a free server being available to subsequent voice arrivals. We further consider admission control along with these policies, to achieve the entire span of the 'achievable region' of ordered pairs of block probability and expected sojourn times. When admission of voice calls is severely reduced, the expected time spent by data calls in the system would be less and vice versa. The two achievable regions overlap, but *PS* has a bigger region (when number of maximum parallel calls is fixed), as it attains a smaller blocking probability. Once the achievable region is known, many relevant optimization problems can be solved easily. For example, the problem of finding the optimal expected sojourn time of data calls, given a constraint on blocking probability of voice calls can readily be solved.

The achievable region is well understood for homogeneous classes, when the performance metric of both the classes is expected sojourn/waiting time (e.g., [2, 5]). Conservation laws, pioneered by [6], capture the fundamental limits of the performance measures like mean waiting time of various classes of customers, when they share a common server. The achievable region has a nice geometric structure (polytopes). While our system has losses

VALUETOOLS 2017, December 5–7, 2017, Venice, Italy

ACM ISBN 978-1-4503-6346-4/17/12…$15.00
https://doi.org/10.1145/3150928.3150953

and hence the conservation laws are not applicable. We conjecture a relationship between the expected sojourn time and the blocking probability (for any static scheduling policy), and, call it a *pseudo conservation law*. This pseudo conservation law is valid only in a fluid limit for short job impatient agents and is proved in [10]. In [10] the performance of the two (*PS* and *CD*) sets of policies is derived (at fluid limit) using the law. While in this paper, we obtain the upper and lower bounds on the performance measures of these two policies (pre limit), and show that the limit of these bounds converge to the same value. We further show that the common limits satisfy the pseudo-conservation law and also achieve all the points of the resulting achievable region. Further we derive the performance of a family of dynamic policies and establish that the dynamic region is strictly bigger.

To the best of our knowledge we are not aware of a work that directly studies this type of a heterogeneous achievable region. In [7] authors consider a multi-class queueing system with eager and tolerant customers. This is the queueing system which is closest to the one considered in this paper, especially to *CD* policy. With our CD policy, the tolerant customers utilize all the remaining servers, and hence the system is work conserving (only) with regard to the tolerant customers. While in [7] tolerant customers are also served in multi-server mode, i.e., each tolerant customer is provided one server and the idle servers (if any) are not utilized. Also, we have more tractable characterization of the performance of the tolerant class, accurate under a fluid limit. We also obtain lower and upper bounds on performance, for the system prior to fluid limit.

There has been considerable work that discusses resource sharing between voice and data calls and we discuss a few here. In [8] authors consider a three channel pool scheme, and obtain a novel adjustable boundary based channel allocation scheme with pre-emptive priority for integrated voice and data networks. They obtain various levels of priority by adjusting the division of the total available channels among the three pools. In [11] authors again consider channel allocation schemes, for packet level allocations. These papers *discuss coarse sharing of resources between data-voice calls.* In our model we provide a scheme to smoothly control the performance measures of the two classes. By varying the admission parameter, smoothly in the interval $[0, 1]$, one can achieve any pair of performance measures on the achievable region.

2 PROBLEM STATEMENT

We refer the impatient/**e**ager customers by ϵ-customers while the **t**olerant customers are referred to as τ-customers. The system has a fixed server capacity, that needs to be shared between the two classes of customers. The exact sharing of capacity depends upon the allocation/scheduling policy. For example the system can serve K customers in parallel for some K, by dividing the server capacity among the customers under service. The system can chose to vary K dynamically, e.g., processor sharing. The system can chose to serve one customer with full capacity etc. In this paper we discuss two example sets of scheduling policies. *We only consider 'τ-work conserving' policies, wherein the τ-customers utilize all the remaining server capacity.*

Arrival process and Jobs: The arrival processes are modelled by independent Poisson processes, with rates λ_ϵ and λ_τ respectively. The job requirements are exponentially distributed. The time required to complete a job, depends upon the scheduling policy. If a τ-customer (ϵ-customer) is served with full capacity, then the service time is exponentially distributed with parameter μ_τ (respectively μ_ϵ).

2.1 Achievable region

The two classes of users have different goals and hence naturally require different qualities of service (QoS). An eager ϵ-agent would leave the facility without service, if service is not provided almost immediately. *Thus Block probability P_B, the probability of customers departing without service,* is an important performance metric for ϵ-class. Further a scheduling decision (basically admission decision) is required at every ϵ-arrival instance[1]. This also implies the service of a typical τ-customer can possibly be interrupted, possibly to provide required QoS for ϵ-agents, and a typical τ-agent can face several such interruptions during its service. Thus the *expected sojourn time $E[S_\tau]$, the expected value of the total time spent by a typical agent would be an appropriate QoS for τ-class.* It is not sufficient to consider the expected waiting time, the time before the service starts. Either of these performance metrics depend upon the scheduler β used. Thus the achievable region is given by:

$$\mathcal{A}_{hetero} = \{(P_B(\beta), E[S_\tau(\beta)]) : \beta \text{ is a scheduler}\}.$$

We begin with *(τ) static policies, wherein the ϵ- admission rules do not depend upon the status of the τ-class.* The probability of admission, p, is an important parameter of any such scheduling policy. Further, the (maximum) number of ϵ-calls served in parallel and the sharing of resources between ϵ and τ customers is also a part of the scheduling decision. For example, the system may allocate/transfer entire server capacity to the first admitted ϵ-arrival. It may processor-share the capacity among the further admitted ϵ-customers. There may be a limit on the number of ϵ-customers that can simultaneously share the capacity. Alternatively the system may allocate a fixed fraction of the server capacity to each admitted ϵ-arrival and the remaining is allocated to τ-class etc. All these rules are independent of the τ-state (e.g., the number of τ customers in the system, waiting time of them etc). This implies that ϵ-calls preempt τ-call when required. In all a τ-static policy implies that an ϵ-arrival is admitted with some probability p, and further admission also depends upon the number of ϵ-calls already in the system, but not on τ-state. Mathematically a static achievable region is defined:

$$\mathcal{A}^{static}_{hetero} = \Big\{ \Big(P_B(\beta^{CS}_p), E[S_\tau(\beta^{CS}_p)]\Big) : \\ 0 \le p \le 1 \text{ and } (CS) \text{ a capacity sharing rule} \Big\}. \quad (1)$$

We primarily analyze the static achievable region. In section 7 we derive the performance of an example family of dynamic (also depends upon τ-state) policies. We also show that the achievable region with dynamic policies is strictly bigger than the static region.

Short-Frequent Job (SFJ) limits. The ϵ-class has short job requirements. If one considers limit $\mu_\epsilon \to \infty$, the impact of ϵ-customers becomes negligible at the limit. To obtain a more general and useful result, we also increase the ϵ-arrival rate while $\mu_\epsilon \to \infty$. That is, every ϵ-agent may utilize the server for a short duration, but the system has to attend the ϵ-agents frequently. Because of this

[1] On the contrary, in homogeneous setting two or more classes of agents wait at their waiting lines and scheduling epochs are the service completion/departure epochs. The scheduler had to decide which class to be served next. While in heterogeneous setting, at any departure epoch there is only one class of agents possibly waiting and hence no decision is required.

ϵ-agents cause significant impact even in the limit. To be more precise we consider the limits $\mu_\epsilon \to \infty$ and $\lambda_\epsilon \to \infty$ while the load factor $\rho_\epsilon = \lambda_\epsilon/\mu_\epsilon$ is maintained constant. We refer this as "Short-Frequent Job (SFJ) limits".

3 PSEUDO CONSERVATION LAW

In a multi-class queueing system with all tolerant classes (homogeneous system), a work conservation law holds. The total workload in the system remains the same irrespective of the scheduling policy, as long as the server does not idle during busy period. Further by Little's law and Wald's lemma, a linear combination of expected sojourn (or waiting) times of different classes of customers remains the same irrespective of the scheduling policy (e.g., [5]).

The above is obviously true when the incoming workload remains the same. However in our heterogeneous setting, the ϵ-customers depart the system, if service is not offered immediately. And this depends upon the scheduling policy. Thus the workload arriving into the system itself changes with different scheduling policies and naturally one may not expect work conservation. However if the amount of work blocked (per unit time) remains the same, one can anticipate a different kind of work conservation. We conjecture that given a probability of blocking, irrespective of the way the ϵ-agents are blocked and irrespective of the way the τ-agents are served, the τ- expected sojourn time remains the same[2]. And this could be conjectured only in SFJ limit and when the policies do not depend upon the τ-state.

In SFJ limit, ϵ-agents will have fluid arrivals and departures. Given the ϵ-load factor (ρ_ϵ) and the probability of blocking (p_B), in the SFJ limit, the ϵ-agents occupy $\rho_\epsilon(1-p_B)$ fraction of system resources at all the times. Hence we *conjecture* that the τ- performance equals that of an $M/M/1$ queue with smaller service rate $\mu_\tau(1-\rho_\epsilon(1-p_B))$, and that the expected sojourn time for any $0 \le p_B \le 1$ equals:

$$E_{S_\tau}(p_B) := \frac{1}{\mu_\tau(1-\rho_\epsilon(1-p_B)) - \lambda_\tau} \text{ if } \rho_\epsilon(1-p_B) + \rho_\tau < 1. \quad (2)$$

Conjecture: Static achievable region, in SFJ limit, equals:

$$\mathcal{A}^{hetero}_{static} = \left\{ \left(p_B, E_{S_\tau}(p_B)\right) \,\middle|\, p_B \in [0,1],\ \rho_\epsilon(1-p_B) + \rho_\tau < 1 \right\}.$$

We would like to refer the equation (2) as a *pseudo conservation law,* as it provides the expected sojourn time in terms of the fraction blocked (lost). This would require an explicit proof which is considered in [10]. For now, we consider two example families of schedulers and illustrate the validity of our conjecture. Further, using the same sets of schedulers, we achieve all the points of the static region. Such a family is generally *referred to as complete family of schedulers.*

4 PROCESSOR SHARING SCHEDULERS

Any ϵ-arrival is admitted to the system with probability p, independent of τ-state. Once admitted it will pre-empt the existing τ-agent, if any. We consider K-processor sharing service discipline for ϵ-agents. If there is only one agent of the ϵ-class receiving service, it is served with maximum capacity, i.e., using capacity μ_ϵ. Upon a new (admitted) arrival of the same class, the capacity is shared among the two. Both are served in parallel and independently, each with rate $\mu_\epsilon/2$. Upon a third (admitted) arrival each is served with rate $\mu_\epsilon/3$. This continues up to K ϵ-agents. Any further arrival, leaves without service even after being admitted. When any of the existing ϵ-agents depart, the service rate is readjusted to an appropriate higher value. The τ-service is resumed only after all the ϵ-agents depart. We call this as $\beta^{PS}_{p,K}$ scheduling policy. Tolerant agents are served in FCFS (first come first serve) basis. They are served in a serial fashion and with full capacity[3] μ_τ. That is, system would serve at maximum one τ-agent, and the service of the next τ-agent begins only after the preceding one departs.

The transitions and evolution of the ϵ-agents is independent of that of τ-agents under a static policy: the arrivals are admitted and the service is provided to the admitted agents immediately, irrespective of the state of τ-agents. Thus one can analyze the ϵ-class independently and we first consider this analysis.

4.1 Blocking Probability of ϵ-class

Fix $0 \le p \le 1$, K and consider policy $\beta^{PS}_{p,K}$. Blocking probability is the probability with which a new (ϵ-class) arrival leaves the system without service. Blocking can occur in case of two events. Upon arrival, an ϵ-agent is admitted to the system with probability p and is blocked with probability $(1-p)$. Secondly, an admitted agent leaves without service, if the system is already serving K ϵ-agents.

Let $\Phi_\epsilon(t)$ represent the number of ϵ-agents in the system at time t. We claim that the ϵ-class transitions are caused by exponential random events and hence that $\Phi_\epsilon(t)$ is a continuous time Markov jump process (see for example [4]) for the following reasons: a) it is clear that the inter-arrival times are exponentially distributed with parameter $\lambda_\epsilon p$; b) it is easy to verify that the departure times are exponentially distributed with parameter μ_ϵ (i.e., $\sim \exp(\mu_\epsilon)$), irrespective of state $\Phi_\epsilon(t)$ (see in [9]).

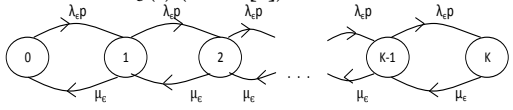

Figure 1: ϵ-State transitions with $\beta^{PS}_{p,K}$ scheduler.

In Figure 1, we depict the transitions of the continuous time Markov jump process $\Phi_\epsilon(t)$. For such processes, well known balance equations are solved to obtain the stationary probabilities (see for example [4]). The stationary probabilities, $\{\pi_0, \pi_1, \cdots, \pi_K\}$, of $\Phi_\epsilon(t)$ are obtained by solving:

$$\pi_0\lambda_\epsilon p = \mu_\epsilon \pi_1,\ \pi_l(\lambda_\epsilon p + \mu_\epsilon) = \lambda_\epsilon p \pi_{l-1} + \mu_\epsilon \pi_{l+1},\ \pi_K \mu_\epsilon = \lambda_\epsilon p \pi_{K-1}.$$

Thus, $\pi_l = \frac{\rho^l_{\epsilon,p}}{a_0}$ with $a_0 := \sum_{j=0}^{K} \rho^j_{\epsilon,p}$ and $\rho_{\epsilon,p} := \frac{\lambda_\epsilon p}{\mu_\epsilon} = \rho_\epsilon p$. An admitted agent gets blocked, if it finds K ϵ-agents in the system, and, this by PASTA (Poisson Arrivals See Time Averages) equals the stationary probability π_K of K ϵ-agents in the system. The agents are not admitted with probability $(1-p)$ and those admitted are blocked with probability π_K. Therefore the overall blocking probability equals:

$$P^{PS}_B(p) = (1-p) + p\pi_K = (1-p) + p\frac{\rho^K_{\epsilon,p}}{a_0}. \quad (3)$$

4.2 Expected sojourn time of τ-class

The ϵ-class requires short but frequent jobs (e.g., voice calls). Hence we are looking for a good relevant approximation that facilitates the

[2]Within tolerant class of customers, the expected sojourn time by Little's law and Wald's lemma is proportional to the workload in the system

[3]Capacity of the server is such that, it can either serve one tolerant agent at rate μ_τ, or l ϵ-agents each at μ_ϵ/l (where $l \le K$).

analysis, and which further allows us to study other important variants (like *CD* policy of section 5). Towards this, we approximately (accurate asymptotically) decouple the evolution of τ-agents from that of ϵ-agents. We first understand the effective server time (EST), Υ_τ, which is defined as the total time period between the service start and the service end of a typical τ-agent. Note that during EST of one agent, no other τ-agent has access to server. Sojourn time of a typical τ-agent equals the sum of two terms: a) waiting time, the time before the service start; and b) EST Υ_τ, the time after the service start.

Effective server time (EST) (Υ_τ). This time equals the sum of the actual service time, B_τ, of the τ-agent and the overall time of interruptions caused by ϵ-agents, which is denoted by Υ^e_τ. Let $N(B_\tau)$ represent the total number of the ϵ-class interruptions, that occurred during the service time B_τ. In reality these interruptions would have occurred in disjoint time intervals, the sum of all of which is B_τ. This random number has same stochastic nature as the number of Poisson arrivals that would have occurred in a continuous time interval of length B_τ. This is true because of the memory less property of the exponential service time B_τ and because Poisson process is a counting process. After an ϵ-agent interrupts the ongoing τ-agent, there is a possibility of further admissions. Eventually the service of the τ-class is resumed, where left, when all the ϵ-agents (that were admitted) leave the system. Thus the time duration for which the service of τ-agent is suspended per interruption, equals a busy period of the ϵ-class, that started with one ϵ-agent. There would be $N(B_\tau)$ (random) number of such interruptions. Hence,

$$\Upsilon_\tau = B_\tau + \Upsilon^e_\tau \text{ with } \Upsilon^e_\tau := \sum_{i=1}^{N(B_\tau)} \Psi_{\epsilon,i}\,, \tag{4}$$

where $\{\Psi_{\epsilon,i}\}_i$ are the IID (independent and identically distributed) copies of ϵ-busy period. We have (proof in Appendix A and [9]):

Lemma 4.1. *The first two moments of the ϵ-busy period and EST Υ_τ are given by:*

$$E[\Psi_\epsilon] = \frac{a_1}{\mu_\epsilon} \text{ and } E[\Psi^2_\epsilon] = \frac{1}{\mu^2_\epsilon}\sum_{i=1}^{K}\frac{q^{i-1}\,c_i}{(1-q)^i}, \tag{5}$$

$$E[\Upsilon_\tau] = \frac{1}{\mu_\tau} + \frac{\lambda_\epsilon p}{\mu_\tau}E[\Psi_\epsilon] \;=\; \frac{a_0}{\mu_\tau}, E[\Upsilon^2_\tau] = \frac{2a_0^2}{\mu^2_\tau} + \frac{\rho_{\epsilon,p}}{\mu_\tau\mu_\epsilon}\sum_{i=1}^{K}\frac{q^{i-1}\,c_i}{(1-q)^i},$$

where the constants q, $\{c_i\}$ and $\{a_i\}$ are defined as:

$$\rho_{\epsilon,p} = \frac{\lambda_\epsilon p}{\mu_\epsilon},\; q = \frac{\rho_{\epsilon,p}}{\rho_{\epsilon,p}+1},\; a_i = \sum_{j=0}^{K-i}\rho^j_{\epsilon,p} \text{ for all } 0 \le i \le K, \tag{6}$$

$$b_i = \sum_{j=K-i+1}^{K-1}(K-j)\rho^j_{\epsilon,p} \text{ for all } 2 \le i \le K, \quad b_1 = 0,$$

$$c_1 = \frac{2\rho_{\epsilon,p}\,(2a_2+b_2)+2}{(1+\rho_{\epsilon,p})^2\mu^2_\epsilon}, \text{ and for all } 1 \le i < K$$

$$c_i = \frac{2\rho_{\epsilon,p}\,((i+1)a_{i+1}+b_{i+1}) + 2\,((i-1)a_{i-1}+b_{i-1})+2}{(1+\rho_{\epsilon,p})^2\mu^2_\epsilon},$$

$$c_K = \frac{2\rho_{\epsilon,p}(Ka_K+b_K)+2((K-1)a_{K-1}+b_{K-1})+2}{(1+\rho_{\epsilon,p})^2\mu^2_\epsilon}. \qquad \blacksquare$$

4.2.1 Approximate decoupling via Domination. Every τ-agent undergoes similar stochastic behaviour, as below. Each agent has to wait for the beginning of its service, and has to finish its service in the midst of random interruptions, all of which have identical stochastic nature. Further, evolution of the ϵ-agents during the EST Υ_τ of one τ-agent is independent of that of the other τ-agents. Hence the Υ_τ times corresponding to different τ-agents are independent of each other. Thus the idea is to model the τ-class evolution approximately as an independent process, with that of an $M/G/1$ queue. The arrivals remain the same, but the service times in $M/G/1$ queue are replaced by the sequence of ESTs $\{\Upsilon^t_\tau\}$.

We call this $M/G/1$ queue as $\mathcal{M}_L$ system and the original system as O system. In fact we will define another $M/G/1$ system $\mathcal{M}_U$ as below and show that: a) the performance (expected sojourn times) of the original system is bounded between the performances of the two $M/G/1$ systems; and b) that the performances of the two sandwiching systems converge towards each other as $\mu_\epsilon \to \infty$ (even with ρ_ϵ fixed).

$\mathcal{M}_L$ system. The ESTs are considered as service times of τ-agent in $\mathcal{M}_L$ system. We study the (sample path wise) time evolution of the two systems, original and $\mathcal{M}_L$, to demonstrate the required domination. Towards this, we assume that both the systems are driven by same input (arrival times and service requirements) processes. Consider that both the systems start with same number (greater than 0) of τ-agents and assume that both of them start with service of the first among the waiting ones. Then the trajectories of both the systems evolve in exactly the same manner, until the τ-queue gets empty. There can be a change in the trajectories of the two systems, upon a subsequent new τ-arrival. We can have two scenarios as in Figure 2. If ϵ-agents are absent at the τ-arrival instance in the original O system (as in sub-figure b), then again, both the systems continue to evolve in the same manner. On the other hand, if ϵ-agents are deriving service (as in sub-Figure a), the service of τ agent is delayed in the original O system till the end of the ongoing ϵ-busy period. While the service starts immediately in $\mathcal{M}_L$ system. Then the trajectories in the two systems continue with the same difference, until the end of the next τ-idle period. At this point the difference: a) either gets reduced, if the τ arrival marking the end of τ-idle period occurs after sufficient time and finds no ϵ-agent; b) or can increase, if the τ-arrival occurs again during an ϵ-busy period; c) or can continue with almost previous value, if the τ-arrival occurs immediately and finds no ϵ-agent. And this continues. Thus the sojourn times in $\mathcal{M}_L$ system are lower than or equal to that in O system in all sample paths. As we notice the difference between the two systems is because of ϵ-busy cycles and this difference may diminish if the later shorten. We will show this indeed is true in coming sections.

$\mathcal{M}_U$ system. Consider another $M/G/1$ system whose service times equal $\Upsilon_\tau + \Psi_\epsilon$, where Ψ_ϵ is an additional ϵ-busy period independent of Υ_τ. It is clear that this system dominates the O system everywhere (see O and $\mathcal{M}_U$ trajectories in Figure 2). Hence the sojourn times of τ-agent in O system are upper bounded by that in $\mathcal{M}_U$ system (in all sample paths). Thus the expected sojourn time of O system is sandwiched as below:

$$E^{\mathcal{M}_L}[S_\tau] \le E^{O}[S_\tau] \le E^{\mathcal{M}_U}[S_\tau]. \tag{7}$$

4.2.2 Performance of $\mathcal{M}_L$ and $\mathcal{M}_U$ systems. In Lemma 4.1, we obtained the first two moments of the ϵ-busy period and the EST, Υ_τ. Using the well known formula for the expected sojourn time of an $M/G/1$ queue, we have:

$$E^{\mathcal{M}_L}[S_\tau] = E[\Upsilon_\tau] + \frac{\lambda_\tau E[\Upsilon^2_\tau]}{2(1-\rho^{\mathcal{M}_L}_\tau)} \text{ with } \rho^{\mathcal{M}_L}_\tau = \lambda_\tau E[\Upsilon_\tau].$$

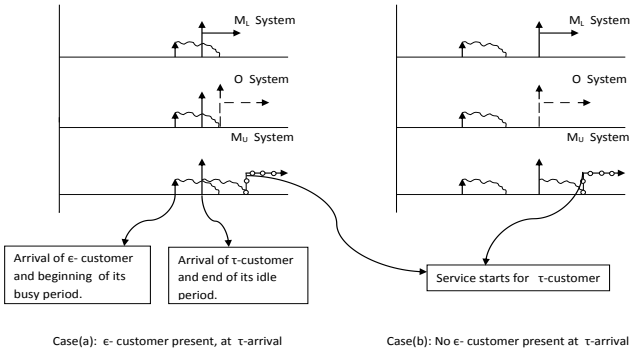

Figure 2: Sample paths of 3 systems (*PS* policy)

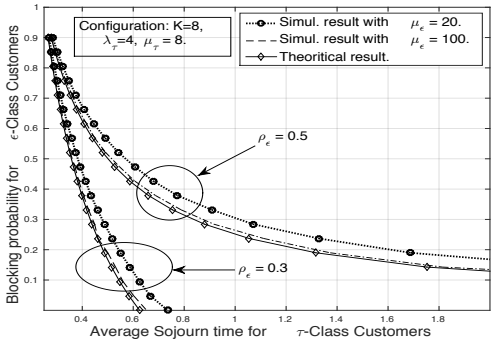

Figure 3: Achievable region

Similarly with $\rho_\tau^{\mathcal{M}_U} = \lambda_\tau E[\Upsilon_\tau + \Psi_\epsilon]$,

$$E^{\mathcal{M}_U}[S_\tau] = E\Big[\Upsilon_\tau + \Psi_\epsilon\Big] + \frac{\lambda_\tau\left(E[\Upsilon_\tau^2] + E[\Psi_\epsilon^2] + 2E[\Psi_\epsilon]E\Upsilon_\tau\right)}{2(1-\rho_\tau^{\mathcal{M}_U})}.$$

From Lemma 4.1 constants $\{c_i\}$, moments of busy period $E[\Psi_\epsilon]$, $E[\Psi_\epsilon^2]$ converge to zero as $\mu_\epsilon \to \infty$, and so the difference $E^{\mathcal{M}_U}[S_\tau]$– $E^{\mathcal{M}_L}[S_\tau]$ converges to zero. In fact this is true even when $\mu_\epsilon, \lambda_\epsilon$ jointly converge to ∞ while maintaining $\rho_\epsilon = \lambda_\epsilon/\mu_\epsilon$ constant. If $\mu_\epsilon \to \infty$ for a fixed λ_ϵ, then the load factor also decreases to zero in limit. Thus the result would have been true only for low load factors. But by maintaining the ratio ρ_ϵ fixed when $\mu_\epsilon \to \infty$, we ensured that *the approximation is good for any given load factor and for any given admission control p, i.e., for any* (ρ_ϵ, p). Under SFJ limit, using Lemma 4.1 and (7):

$$E_{PS}[S_\tau(p)] \quad := \quad E^{O}_{PS}[S_\tau(p)] \approx \frac{1}{\tilde{\mu}_{\tau,p}(1-\tilde{\rho}_{\tau,p})}, \tag{8}$$

$$\text{with } \tilde{\rho}_{\tau,p} \quad = \quad \rho_\tau a_0, \;\; \tilde{\mu}_{\tau,p} = \frac{\mu_\tau}{a_0} \text{ and } \rho_\tau := \frac{\lambda_\tau}{\mu_\tau}.$$

Thus the achievable region under SFJ limit is given by:

$$\mathcal{A}_{PS} = \left\{ \left((1-p) + \frac{p(\rho_{\epsilon,p})^K}{a_0}, \;\; \frac{a_0}{\mu_\tau\,(1-a_0\rho_\tau)} \right) \;\middle|\; a_0\rho_\tau < 1, \; 0 \le p \le 1 \right\}.$$

In the above, condition $a_0\rho_\tau < 1$ ensures stability.

Validation-Pseudo conservation law (2), Completeness. By direct substitution[4] one can verify that the performance measures of $\beta^{PS}_{p,K}$ scheduler, for every (p,K), satisfy the pseudo conservation law (2). Further as K increases to ∞, the blocking probability $P_B^{PS}(1)$, given by equation (3), decreases to zero if $\rho_\epsilon \le 1$. When $\rho_\epsilon > 1$, using simple computations[5], one can show that $P_B^{PS}(1) \to 1 - 1/\rho_\epsilon$ and only $p_B > 1 - 1/\rho_\epsilon$ can be a part of the $\mathcal{A}^{static}_{hetero}$. Also it is easy to verify that the function, $p \mapsto P_B^{PS}(p)$, is continuous in p for any K. Thus by intermediate value theorem, all the points of $\mathcal{A}^{hetero}_{static}$ can be achieved by these schedulers. And hence the family of schedulers,

$$\mathcal{F}^{PS} := \left\{ \beta^{PS}_{p,K}, \; 0 \le p \le 1, K \right\},$$

is complete, when $\rho_\epsilon \le 1$. It is important to note here that these schedulers achieve the entire static region, nevertheless a larger K implies a larger time spent by ϵ-agents in the system. Thus system may have a restriction on the size of K to be used based on other QoS requirements.

5 CAPACITY DIVISION (*CD*) POLICIES

In the previous section, when an admitted ϵ-customer pre-empts the ongoing service of τ-customer the entire system capacity is transferred to ϵ-customer. In this section we analyze a different scheduling policy. Here the capacity is not completely transferred, but rather a fraction of it is used by each ϵ-customer. The τ-customer is continued with the remaining capacity.

Each ϵ-customer uses $(1/K)$-th part of the capacity, μ_ϵ/K. If the system has only one ϵ-customer, the remaining capacity i.e., $(K-1)/K$-th part of the capacity is utilized by the τ-customer. In other words, τ-class is served with rate $\mu_\tau(K-1)/K$. If there are $0 \le l \le K$ number of ϵ-customers receiving the service, then (l/K)-th part of the capacity is used by the ϵ-customers and the τ-customer is served at rate $((K-l)/K)\mu_\tau$. This continues up to K ϵ-customers, and any further (admitted) ϵ-arrival departs without service. Whenever an existing ϵ-customer departs, the capacity is *readjusted to an appropriate higher value for τ-customer.*

The block probability equals exactly that of an $M/M/K/K$ queue (see in [9]). We now discuss the expected sojourn time and more details are in [9]. The idea is once again to approximately decouple the evolution of τ-agents from that of ϵ-agents. Once again EST is denoted as $\check{\Upsilon}_\tau$, has similar meaning as in section (4.2) and typical τ-sojourn time equals the sum of waiting time and the effective server time (EST), $(\check{\Upsilon}_\tau)$. However the EST now depends upon the number of ϵ-agents in the system at the service start and hence the analysis is more complicated. Let $\check{\Upsilon}^l_\tau$ represents the EST, when it starts with l ϵ-agents. In [9] we derived the moments of $\check{\Upsilon}^l_\tau$ and these provide the upper and lower bounds of the performance even before the fluid limit. In particular (proof in [9]).

Theorem 1. *The first two moments of EST $\check{\Upsilon}^0_\tau$ ($l = 0$) are:*

$$E[\check{\Upsilon}^0_\tau] \quad = \quad \frac{\check{a}_0 + O(1/\mu_\epsilon)}{\eta\mu_\tau + O(1/\mu_\epsilon)}, \;\; \check{a}_0 := \sum_{j=0}^{K} \frac{\left(\rho_{\epsilon,p}\right)^j}{j!} \text{ and} \tag{9}$$

$$E\Big[\left(\check{\Upsilon}^0_\tau\right)^2\Big] \quad = \quad \frac{2\frac{\check{a}_0^2}{\eta\mu_\tau} + O(1/\mu_\epsilon)}{\eta\mu_\tau + O(1/\mu_\epsilon)} \text{ with } \eta := \sum_{j=0}^{K-1} \frac{\left(\rho_{\epsilon,p}\right)^j}{j!}\frac{K-j}{K},$$

where $f(\mu_\epsilon) = O(1/\mu_\epsilon)$ for any function f implies, $f(\mu_\epsilon)\mu_\epsilon \to$ constant as $\mu_\epsilon \to \infty$, with ρ_ϵ fixed. ∎

[4] By (3), $1-\rho_\epsilon(1-P_B^{PS}) = 1/a_0$ and so $\left(\mu_\tau\left[1-\rho_\epsilon(1-P_B^{PS})\right]-\lambda_\tau\right)^{-1}$ (see equation (2)) equals $E_{PS}[S_\tau]$ given by (8).

[5] It is easy to verify as $K \to \infty$ that:

$$\frac{\rho_\epsilon^K}{\sum_{l=0}^K \rho_\epsilon^l} = \frac{1}{\sum_{l=0}^K \rho_\epsilon^{-(K-l)}} = \frac{1}{\sum_{l=0}^K \rho_\epsilon^{-l}} \to \frac{1}{\frac{1}{1-\rho_\epsilon^{-1}}} = 1 - \frac{1}{\rho_\epsilon}.$$

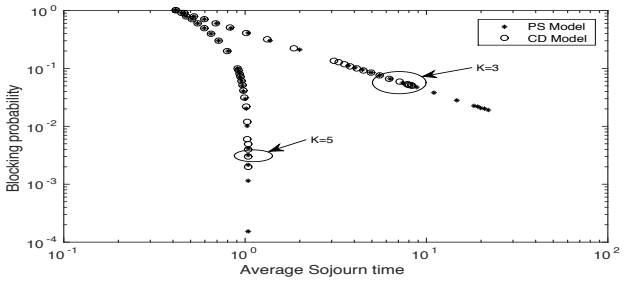

Figure 4: Achievable regions $\mathcal{A}_{CD}$, $\mathcal{A}_{PS}$, with $\rho_\epsilon = 0.9/K$

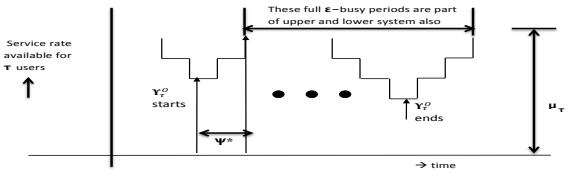

Figure 7: Effective server time, $\check{\Upsilon}_\tau$, in *CD* Model)

Dominating systems. It was not difficult to obtain the conditional moments of the EST, $\{\check{\Upsilon}_\tau^l\}_l$. However to obtain the unconditional moments, one requires the stationary distribution of the ϵ-number l at service start of a typical τ-agent. And this is not a very easy task. However the various conditional moments differ from each other (mostly) at maximum in one ϵ-busy period (see Figure 7). Hence one can possibly obtain the (approximate) unconditional moments, along with M/G/1 queue approximation, using the idea of dominating fictitious queues. This is taken up immediately.

We again construct two dominating systems, whose IID service times 'dominate' either side of the sequence $\{\check{\Upsilon}_{\tau,n}^{l_n}\}_n$. We first discuss the upper bounding system. The service times in original system $\check{\Upsilon}_{\tau,n}^{l_n}$ (for any n) can start and end in between ϵ-busy period(s) as in Figure 7. Further these residual busy periods are correlated, for example the starting residual ϵ-busy period (call this as Ψ^*) is correlated with $\check{\Upsilon}_{\tau,n-1}^{l_{n-1}}$ of previous τ-customer. To dominate any $\check{\Upsilon}_\tau^l$ of original system with an IID version, we first replace Ψ^* with busy period $\tilde{\Psi}^*$ of a CD system with $2K$ servers (each of capacity μ_ϵ/K), when started with K ϵ-customers and such that: a) if l number of ϵ-customers are deriving service at the beginning of Ψ^*, the residual service times (which are again exponential with parameter μ_ϵ/K, because of memoryless property) of those l customers also equal the service time requirements of the first l customers of the $2K$ system; b) the service times of the remaining $(K-l)$ ϵ-customers are independent copies of the exponential random variable with the same parameter; c) further inter arrival times and service times of all the new ϵ-customers coincide with that in the original system; and d) if a customer is not accepted in original system, we consider an independent service time for that customer. With this construction, an ϵ-customer departure during Ψ^* of the original system definitely marks a departure in $2K$ system also, any customer accepted in original system is also accepted in the $2K$ system. Thus the busy period $\tilde{\Psi}^*$ of the $2K$ system dominates the residual ϵ-busy period Ψ^* at the start of the τ-customer service, irrespective of the number, l, of ϵ-customers existing in the original system at the start of Ψ^*. In other words, this time (corresponding to n-th user) is independent

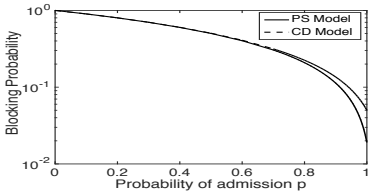

Figure 5: P_B versus p

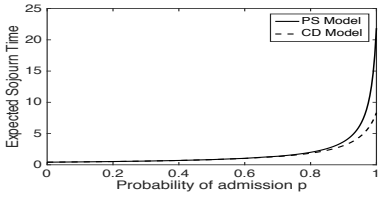

Figure 6: $E[S_\tau]$ versus p

of the quantities related to all other ($\neq n$) τ-customers (original system) and dominates Ψ^* of the n-th customer almost surely.

The above constructed $\tilde{\Psi}^*$ of the $2K$ system forms the beginning part of the n-th τ-customer service time in upper system with following additional details: a) the τ-customer in upper system is not served at the beginning for a duration equal to $\tilde{\Psi}^*$; b) the service of τ-customer in upper system starts with full ϵ-busy periods, and we assume these equal the full ϵ-busy periods of the original system that interrupted the n-th τ-customer's service; c) if extra ϵ-busy periods are required to complete the τ-job we add independent copies of the ϵ-busy periods, but we do not couple the ϵ-busy periods that interrupt the $(n+1)$-th τ-customer. Clearly τ-customer spends more time in upper system than in the original system.

A lower dominating system is obtained by using exactly the same construction, but here the τ-customers are served with full capacity during $\tilde{\Psi}^*$, constructed using the $2K$ system. Thus clearly the τ-customers spend less time (almost surely) in the lower system. And further the difference between the two dominating systems converges to zero because $\tilde{\Psi}^*$ (the busy period of CD system with $2K$ servers) also converges to zero as in proof of Theorem 1 (in [9]).

Performance. The expected sojourn time of the *CD* model can also be obtained as limit of the expected sojourn times of M/G/1 queues with service time moments given by that of $\check{\Upsilon}_\tau^0$ of Theorem 1. Thus the achievable region in the SFJ limit is given by ([9]):

$$\mathcal{A}_{CD} = \left\{ \left((1-p) + p\frac{(K\rho_{\epsilon,p})^K}{K!\check{a}_0},\ \frac{1}{\ddot{\mu}_{\tau,p}\left(1-\ddot{\rho}_{\tau,p}\right)} \right) : \ddot{\rho}_{\tau,p} < 1,\ 0 \le p \le 1 \right\}, \text{ with } \ddot{\rho}_{\tau,p} = \frac{\lambda_\tau}{\ddot{\mu}_{\tau,p}},$$

$$\check{a}_0 := \sum_{j=0}^{K} \frac{\left(K\rho_{\epsilon,p}\right)^j}{j!},\ \eta := \sum_{j=0}^{K-1} \frac{\left(K\rho_{\epsilon,p}\right)^j}{j!}\frac{K-j}{K},\ \text{and } \ddot{\mu}_{\tau,p} = \frac{\eta\mu_\tau}{\check{a}_0}.$$

By direct substitution, we see that the *CD* policies also satisfy the pseudo conservation law (2). They also form a complete family of schedulers, for exactly the same reasons as that for *PS* policy when $\rho_\epsilon \le 1$ (details in [9]). For $\rho_\epsilon > 1$, they cover only partial achievable region ([9]).

6 NUMERICAL EXAMPLES

We conduct Monte-Carlo simulations to estimate the performance of both the policies. We basically generate random trajectories of the two arrival processes, job requirements and study the system evolution when it schedules agents according to *PS*/*CD* policy. We estimated the blocking probability and expected sojourn time for ϵ and τ-agents respectively, using sample means, for different values of (p, K).

In Figure 3, we consider an example to compare the theoretical expressions with the ones estimated using Monte-Carlo simulations for *PS* policy. We consider two different values of ρ_ϵ. We notice negligible difference between the theoretical and simulated values with $\mu_\epsilon = 100$. However even with $\mu_\epsilon = 20$, the difference is about 10-12% for most of the cases. Some more examples, including *CD* policy, are in [9].

We compare the achievable regions of *PS* and *CD* policies by plotting $\mathcal{A}_{CD}$ and $\mathcal{A}_{PS}$. We set $\rho_\epsilon = 0.9/K$, $\lambda_\tau = 5.6$ $\mu_\tau = 8$ and $K = 3$ or 5. In Figure 4, we plot the achievable region for both the models/policies, i.e, we plot $E[S_\tau(p)]$ versus $P_B(p)$, for different p. And in Figures 5 and 6, we plot the performance measures $P_B(p)$ and $E[S_\tau(p)]$ respectively versus p with $K = 3$. From Figure 4, the two achievable (sub) regions overlap, however we observe from the Figures 5 and 6 that the performance measures of the two models are different for the same (p, K). But if we choose a p and $p\prime$ such that $P_B^{CD}(p) = P_B^{PS}(p\prime)$, we observe that the two expected sojourn times are equal. Because of this the two achievable regions overlap in Figure 4. This observation *is precisely the pseudo-conservation law (2). Whatever the policy used, once the blocking probabilities are the same, the expected sojourn times are also the same.*

Now we will discuss a slightly different, yet, a related important aspect. We would compare the two sets of policies, when K (maximum number of parallel ϵ-calls) is the same. As seen from the figures the sub-achievable region of *CD* policy, with fixed K, is a strict subset of that of the *PS* policy. This is because the best possible blocking probability with *CD* policy,

$$P_B^{CD}(1) = \frac{(K\rho_\epsilon)^K/K!}{\sum_{j=0}^{K}(K\rho_\epsilon)^j/j!} \geq \frac{(\rho_\epsilon)^K}{\sum_{j=0}^{K}(\rho_\epsilon)^j} = P_B^{PS}(1),$$

is greater than that with the *PS* policy. In Figure 4 the best P_B with *CD* and *PS* models/policies respectively is 0.002 and 0.0002 (0.05 and 0.019) when $K = 5$ ($K = 3$). Thus it appears that the static achievable region would overlap for different policies, however the sub-regions covered by different policies can be different when K is fixed. By increasing K, one can achieve the entire static region and those examples are considered in [9].

7 A DYNAMIC FAMILY OF SCHEDULERS

We consider dynamic policies (for *PS* model) with an aim to demonstrate that the dynamic region is bigger than the static region. Towards this we construct an example dynamic policy and show that the block probability, for the same sojourn time $\mathbb{E}[S_\tau]$, is better with the dynamic policy.

The static policy of the previous sections is modified as follows. We refer this as policy β_p^d. When there are no τ-agents in the system, i.e., during the τ-idle period, there is no admission control for ϵ-agents. An arriving ϵ-agent is admitted with probability one. Recall, however that service is offered to an admitted agent only when

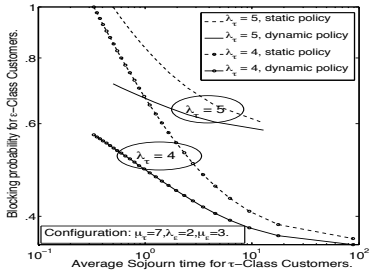

Figure 8: Static-dynamic policies, $K = 4$**.**

the number in system is less than K. When the system is in τ-busy period[6], i.e., when the τ-queue is non-empty, we admit the ϵ-agents with probability p. So, this is a dynamic policy which alternates between full and partial admission.

Let Ψ_τ and $\mathcal{I}_\tau$ respectively represent the busy and idle periods of the τ-agents. By stationarity, memoryless property, the consecutive busy, idle periods $\{\Psi_{\tau,i}\}$, $\{\mathcal{I}_{\tau,i}\}_i$ are independent and identically distributed. We have (proof in [9]):

Theorem 2. *The block probability,* $P_d^B(p)$*, for the system with the dynamic policy* β_p^d*:*

$$P_d^B(p) = \frac{\mathbb{E}[\mathcal{I}_{\tau,1}]P^B(1)}{\mathbb{E}[\Psi_{\tau,1}] + \mathbb{E}[\mathcal{I}_{\tau,1}]} + \frac{\mathbb{E}[\Psi_{\tau,1}]P^B(p)}{\mathbb{E}[\Psi_{\tau,1}] + \mathbb{E}[\mathcal{I}_{\tau,1}]}. \quad \blacksquare \quad (10)$$

Using the ideas of dominating systems as in the section 4 one can show that the moments of the idle, busy periods of the original system with policy β_p^d converges towards that of the equivalent $M/G/1$ system $\mathcal{M}_L$, as $\mu_\epsilon \to \infty$. Thus we will have for large values of μ_ϵ:

$$\mathbb{E}[\mathcal{I}_{\tau,1}] \approx \frac{1}{\lambda_\tau},\ \mathbb{E}^O[\Psi_{\tau,1}] \approx \mathbb{E}^{\mathcal{M}_L}[\Psi_{\tau,1}] = \frac{\mathbb{E}[\Upsilon_\tau]}{1-\lambda_\tau\mathbb{E}[\Upsilon_\tau]} \to \frac{a_0}{\mu_\tau - \lambda_\tau a_0}.$$

The second last equality is obtained using the well known formula for the average busy period of an $M/G/1$ queue. It is easy to see that the sojourn time of the dynamic policy β_p^d is same as that with static policy β_p (asymptotically), while the blocking probability is improved from (3) to (10). Note that $P^B(1) \leq P^B(p)$ for any $p \leq 1$. Hence the dynamic policy performs better and the dynamic achievable region is bigger. Similar improvement is possible with *CD* policy.

Numerical comparison

In Figure 8, we compare the performance of the dynamic policy β_p^d with the corresponding static policy, for *PS* model. We notice a good improvement in the curve: blocking probability decreases significantly for the same expected sojourn time. This indicates that the dynamic region is strictly bigger than the static region, unlike the homogeneous case. In future, we would like to obtain complete analysis of dynamic achievable region for this heterogeneous system.

8 CONCLUSIONS

We consider a queueing system with heterogeneous classes of agents. The impatient class demands immediate service, hence

[6]Normally a busy period begins immediately with an arrival to an empty queue. However, in our system we say a τ-busy period starts with the service start of that τ-agent, which arrives to an τ-empty queue. If ϵ-agents were present at the τ-arrival instance, the service of the τ-agent is deferred till the end of the ongoing ϵ-busy period.

receives the service immediately and if required in parallel with others. There is an admission control to ensure the QoS requirements of the other (tolerant) class. The tolerant class can wait for their turn, however would like to optimize their sojourn time.

We conjecture a pseudo conservation law for this lossy queueing system, which relates the blocking probability of impatient agents to the expected sojourn time of the tolerant agents, in a short and frequent job (SFJ) limit-regime for the former. This law should be satisfied by all the policies, that are static (do not depend on τ-state) and work conserving (left over server capacity is completely used when there is a customer) with respect to the tolerant agents.

We consider two families of scheduling policies, which differ in the way the system capacity is shared between the two classes. With processor sharing policy the entire system capacity is transferred to (admitted) impatient customers. In the second policy, which we refer to as capacity division policy, only a (fixed) fraction of the capacity is transferred to each admitted impatient customer.

We obtain closed form expressions for the asymptotic performance measures, under SFJ limit, for both the families. The two families satisfy the pseudo-conservation law. Further, both the families are complete, i.e., they attain every point of the achievable region given by the pseudo-conservation law. The *CD* achievable region is a strict subset of the *PS* region, when restricted to the same number of parallel service possibilities. This demonstrates the limitation of the *CD* policy, which could be a more practically used model.

The results are asymptotic and are accurate when the arrival-departure rates of the impatient class is large. Usually such customers have short frequent job requirements and hence this is an useful asymptotic result. Further, we have an upper and lower bound for the sojourn time performance, even when the rates are not large.

We also consider an example family of dynamic policies, derive their performance and establish that the dynamic region is strictly bigger than the static region. This is in contrast with homogeneous (all tolerant classes) system, where dynamic and static regions coincide. Consider a static policy for homogeneous system which schedules (after every departure) the server to one of the two tolerant classes with probability p independent of the system state. One can easily show that these policies (as p varies in $[0,1]$) achieve the entire region.

REFERENCES

[1] Sesia, S., Baker, M., and Toufik, I.'LTE-the UMTS long term evolution: from theory to practice', *John Wiley & Sons*, 2011.

[2] E. G. Coffman and I. Mitrani, 'A characterization of waiting time performance realizable by single server queues', *Operations Research*, vol. 28, 1979.

[3] D. Bertsimas, I. Paschalidis, and J. N. Tistsiklis, 'Optimization of multiclass queueing networks: Polyhedral and nonlinear characterizations of achievable performance', *The Annals of Applied Probability*, 1994.

[4] Hoel, Paul G., Sidney C. Port, and Charles J. Stone. 'Introduction to stochastic processes', 1986.

[5] J. G. Shanthikumar and D. D. Yao, 'Multiclass queueing systems: Polymatroidal structure and optimal scheduling control', *Operations Research*, vol. 40, no. 3-supplement-2, pp. S293–S299, 1992.

[6] L. Kleinrock, 'A conservation law for wide class of queue disciplines', *Naval Research Logistics Quarterly*, vol. 12, June-September 1965.

[7] Sleptchenko, A., A. van Harten, and M. C. van der Heijden. 'An Exact Analysis of the Multi-class M/M/k Priority Queue with Partial Blocking', pp. 527-548, 2003.

[8] S. Tang and Wei Li, 'A Channel Allocation Model with Preemptive Priority for Integrated Voice/Data Mobile Networks', *Proceedings of the First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, IEEE*, 2004.

[9] Veeraruna Kavitha and Raman Kumar Sinha, 'Queuing with Heterogeneous Users: Block Probability and Sojourn times', arXiv preprint arXiv:1709.06593 (2017).

[10] Veeraruna Kavitha, Jayakrishnan Nair and Raman Kumar Sinha, 'Pseudo conservation for partially fluid, partially lossy queueing systems', accepted with minor revision, Annals of Operations Research.

[11] Yan Zhang, Boon-Hee Soong and Miao Ma, 'A dynamic channel assignment scheme for voice/data integration in GPRS networks', *Elsevier Computer communications*, 29, pp. 1163–1173, 2006.

APPENDIX A: PROOF OF LEMMA 4.1

By conditioning on B_τ, one can verify that

$$E[N(B_\tau)] = \frac{\lambda_\epsilon p}{\mu_\tau},\ E[B_\tau N(B_\tau)] = \frac{2\lambda_\epsilon p}{\mu_\tau^2},$$
$$E[(N(B_\tau))^2] = \frac{\lambda_\epsilon p}{\mu_\tau} + \frac{2(\lambda_\epsilon p)^2}{\mu_\tau^2}.$$

By conditioning on $N(B_\tau)$ we obtain the first moment:

$$E[\Upsilon_\tau] = E[B_\tau] + E\left[\sum_{i=1}^{N(B_\tau)} \Psi_{\epsilon,i}\right] \tag{11}$$
$$= E[B_\tau] + E\left[E\left[\sum_{i=1}^{N(B_\tau)} \Psi_{\epsilon,i}\middle| N(B_\tau)\right]\right] = \frac{1}{\mu_\tau} + \frac{\lambda_\epsilon p E[\Psi_\epsilon]}{\mu_\tau}.$$

Note that the busy periods $\{\Psi_{\epsilon,i}\}_i$ are IID. From (4) we have:

$$E[\Upsilon_\tau^2] = E[B_\tau^2] + 2E\left[B_\tau \Upsilon_\tau^e\right] + E\left[(\Upsilon_\tau^e)^2\right]. \tag{12}$$

By first conditioning on $(B_\tau, N(B_\tau))$ and then on B_τ:

$$E\left[B_\tau \Upsilon_\tau^e\right] = E\left[E\left[B_\tau \sum_{i=1}^{N(B_\tau)} \Psi_{\epsilon,i}\middle| B_\tau, N(B_\tau)\right]\right]$$
$$= \lambda_\epsilon p E\left[\Psi_\epsilon\right] E\left[B_\tau B_\tau\right] = \frac{2\lambda_\epsilon p E\left[\Psi_\epsilon\right]}{\mu_\tau^2}. \tag{13}$$

Conditioning as before and because of independence:

$$E\left[(\Upsilon_\tau^e)^2\right] = E\left[E\left[\left(\sum_{i=1}^{N(B_\tau)} \Psi_{\epsilon,i}\right)^2\middle| N(B_\tau)\right]\right],$$
$$= \frac{\lambda_\epsilon p E\left[\Psi_\epsilon^2\right]}{\mu_\tau} + \frac{2(\lambda_\epsilon p)^2}{\mu_\tau^2}\left(E\left[\Psi_\epsilon\right]\right)^2, \text{ which simplifies to (5).}$$

Busy period of ϵ-class. Busy period of any class is defined as the time till the first epoch at which all the customers of that class have departed. Let Ψ_k, represent the busy period of ϵ-class, when it begins with k number of customers. Note that $\Psi_\epsilon = \Psi_1$. In all the discussions below, an arrival is meant an admitted arrival.

The busy period Ψ_1 starts with the arrival of one ϵ-customer. If the customer leaves before the next arrival, the busy period ends. On the other hand, if an arrival occurs before the departure of the existing customer, it marks the beginning of a busy period with two customers, Ψ_2. As seen in section 4.1 (see Fig. 1), a departure time is memoryless, i.e., exponential random variable with parameter μ_ϵ irrespective of the number of customer sharing the service. Let D represent the departure time. The inter arrival time, A, is exponential with parameter $\lambda_\epsilon p$. Let $W := \min\{D, A\}$ represent the minimum of the two. With these definitions:

$$\Psi_1 = 1_{\{D<A\}}\, 0 + 1_{\{A<D\}}\, \Psi_2 + W. \tag{14}$$

One can write similar equations for Ψ_k, busy period starting k-ϵ-customers and complete the proof by computing their expected values using backward recursion (details in [9]). ■