Performance Simulation and Analysis for LTESystem Using Human Behavior Queue Model

Tony Tsang

Hong Kong Polytechnic University
Hung Hom, Hong Kong.
Email: ttsang@ieee.org

Abstract

Understanding the nature of traffic has been a key concern of the researchers particularly over the last two decades and it has been noticed through extensive high quality studies that traffic found in different kinds of IP/wireless IP networks is human operators . Despite the recent findings of real time human behavior in measured traffic from data networks, much of the current understanding of IP traffic modeling is still based on simplistic probability distributed traffic. Unlike most existing studies that are primarily based on simplistic probabilistic model and traditional scheduling algorithms, this research presents an analytical performance model for real time human behavior queue systems with intelligent task management traffic input scheduled by a novel and promising scheduling mechanism for 4G-LTE system. Our proposed model is substantiated on human behavior queuing system that considers real time of traffic exhibiting homogeneous tasks characteristics. We analyze the model on the basis of newly proposed scheduling scheme for 4G-LTE system. We present closed form expressions of expected response times for real time traffic classes. We develop a discrete event simulator to understand the behavior of real time of arriving tasks traffic under this newly proposed scheduling mechanism for 4G-LTE system . The results indicate that our proposed scheduling algorithm provides preferential treatment to real-time applications such as voice and video but not to that extent that data applications are starving for bandwidth and outperforms all other scheduling schemes that are available in the market.

1. INTRODUCTION

In the Internet, Quality of Service (QoS) management allows different types of traffic to contend inequitably fornetwork resources. Bandwidth is the key heuristic to manage real life network utilities like video and voice overremote locations. Three main QoS frameworks such as IntServ, DiffServ and MPLS have been introduced to providedifferential treatment to a variety of applications available in real time service internet [1] . The differentiation ofmultiple classes of traffic is fundamentally relied on these frameworks that utilize various queuing and schedulingcombinations for separating different traffic classes. Further, the traffic separation is categorized under specificparameters like bandwidth, delay, jitter and packet-loss rate. The different arrangements of these parameters can bebundled under variety of queuing and scheduling methods. It is therefore vital to QoS frameworks that modelingof traffic behavior through network domains is accurate so that resources can be optimally assigned.

Understanding the nature of traffic has been a key concern of the researchers particularly over the last twodecades and it has been noticed through extensive high quality studies that traffic found in different kinds ofIP/wireless IP networks is human operators [2] . Despite the

DOI : 10.5121/ijwmn.2013.5208

recent findings of real time human behavior inmeasured traffic from data networks, much of the current understanding of IP traffic modeling is still based onsimplistic Poisson distributed traffic. In this paper, we add to a more realistic modeling of network domains throughthe following main contributions: (1) the presentation of an analytical approach and closed form expressions tomodel the accurate behavior of multiple classes of wireless IP traffic based on a human behavior queuing systemunder real time assumptions, (2) the derivation of expected waiting times of corresponding human behavior trafficclasses and formulation of an embedded intelligent task management and (3) the detailed simulation results to giveexact QoS parameter bounds to validate the analytical framework.

We have analyzed the traditional scheduling schemes based on real time human behavior queuing system, whereasin current study, we analyze the newly proposed scheduling scheme to guarantee tight bound QoS to all kind of trafficin human behavior service wireless internet. The rest of the paper is structured as follows. Section II summarizesrelated work. The Real Time Human Behavior Queue Model have been discussed in Section III. The simulationand analysis results are given in Section IV. Finally, Section V concludes this work.

2. RELATED WORK

Queuing theory is the backbone of telecommunication systems. The major concern about internet traffic is: howburstiness (commonly known as human behavior operations) behavior can be managed in real time spans. Theexperimental queuing analysis and simulation studies with human behavior packet data arrival traffic have beenperformed in [3] and [4] respectively. These studies merely indicate that providing hard and tight bound guaranteesfor different QoS parameters such as maximum delay, delay-jitter and cell-loss probabilities in the presence ofhuman behavior traffic is nontrivial especially if the coefficient of variation of the marginal distribution is large.The readers are referred to [5, 6] to get a detailed overview of other queuing based results available in the presenceof human behavior traffic. The core limitation of these findings is based on the fact, that FIFO logic has beenconsidered to understand the behavior of traffic, which can’t be used to provide differential treatment to multipleclasses of traffic with different QoS time constraints.

The desire to dispense divergent QoS guarantees to different classes of customers in wireless Internet is leadingto the use of priorities in tenns of allocation of resources. Multiple priority based classes are supported by the IProuters and ATM switches. The authors in [7] have used human-in-the-loop (analytical) Model to provide numericalresults for two different classes of traffic input based on Real Time Task Release Control Process (RT-TRCP). Anotable discrepancy of RT-TRCP is the estimation of large set of parameters. It has been shown that Real Time TaskRelease Control Process (RT-TRCP) can prioritize each class in its own buffer [8]. The flow control managementbased on the computation of probability of various types of traffic classes has been discussed in [9]. The otherwork related to this study can be found in studies [10, 11] . Unfortunately, in related work, the issue of providingQoS guarantees to the end-user based on tight bound QoS parameters has not been properly addressed.

In addition, we refer the readers to [12, 13] regarding the work that has been carried out in terms of IP networkperformance evaluation. The analysis conducted in [12, 13] has two main disadvantages; first the reported queuingmodels did not employ human-in-the-loop phenomenon for network traffic input and second, they have only usedsingle class of traffic for conducting analysis by neglecting the performance affect of other subsequent traffic classes.To

overcome the limitations of related work, they presented a novel analytical framework [14, 15] based on realtime human behavior queuing system, that contemplates task management of human behavior traffic. In the relatedwork, they analyzed the traditional scheduling schemes such as priority and round robin. It is well known thattraditional scheduling schemes can't provide the required QoS to all types of traffic found in modem wirelessnetworks. Hence, in this current study, we analyze real time human behavior model on the basis of a novel andmost promising scheduling mechanism titled as, "Best Scheduling Algorithm (BSA)" and find exact packet delaysfor the corresponding classes of human behavior traffic. The results indicate that BSA completely outperforms alltraditional and other available scheduling schemes. To date, no closed form expressions have been presented forreal time human behavior model with such scheduling mechanism.

3. REAL TIME HUMAN BEHAVIOR QUEUE MODEL

The basic elements of Real Time Human Behavior Queue Model are its actions, which represent activities carriedout by the systems being modeled, and its operators, which are used to real time descriptions.

Time point

A time point is a time instant with respect to the global clock of the system; it does not have duration. It specifiesthe starting and stopping times of an action. Using a time point, we can instruct the system to generate an actionat a particular point in time. Time point progresses consistently in all parts of the system. More formally, the timepoint is defined by using a discrete time domain, which contains the following properties:

$$\forall t \forall t' t < t' \rightarrow \forall t'' \rightarrow t < t'' \vee t' \leq t''$$

We assume a fixed set of clocks $t = \{t_0, \dots, t_i\}$. The special time point t_0, which is called the start time point, always has the value 0.

Time Constraint

An action can exist for a period of time; this duration is called the time constraint of the action. A time constrainthas a starting and an ending point. It consists of a lower-bound and an upper-bound time point, where the lowerboundtime point enables an action in a module, and the upper-bound time point disables the action at that pointin time. Formally, we define a time constraint in the following:

$$\mathcal{T}_i = \{[\tau_{i_{min}}, \tau_{i_{max}}] \mid \forall t_i \in T\} \text{with } 0 \leq \tau_{i_{min}} \leq \tau_{i_{max}}.$$

Timed Action

A timed action is a tuple $< \alpha, \lambda, \mathcal{T} >$consisting of the type of the action α, the rate of the action $\lambda$and temporalconstraint of the action $\mathcal{T}$. The type denotes the kind of action, such as transmission of data packets, while therate indicates the speed at which the action occurs from the view of an external observer. The rates are used todenote the random variables specifying the duration of the actions. The actions can be defined in different types ofprobability distribution function such as human behaviors distribution. Moreover, each transition is also boundedby a temporal constraint.

Real Time Single Server Queue Model

Consider the following single-server queue model. Tasks arrive periodically, at rate λ, i.e., a new task arrives every $1/\lambda$ time units. The tasks are identical and independent of each other and each task brings w units of work, where w is an independent identically distributed (i.i.d.) random variable whose probability distribution is f_W with bounded support $[\mathcal{W}_1, \mathcal{W}_2]$ for some $\mathcal{W}_1 > 0$ and $\mathcal{W}_1 \leq \mathcal{W}_2$. In the rest of the paper, we will assume this bounded support assumption on f_W without explicitly repeating it. Let w be the mean of w with respect to f_W. Let δ_w be the Dirac delta distribution centered at $\bar{w}$. We will use the δ distribution for the scenario when the tasks are homogeneous. Note that the task arrival process under consideration is deterministic. We briefly discuss the implications of stochastic inter arrival times in Section. The tasks must be serviced in the order of their arrival. We next state the dynamical model for the server, which specifies the state-dependent service times for the server.

3.1 Real Time Server Model

Let $x(t) \in [0,1]$ be the server state at time t, and let $b: \mathcal{R} \to \{0,1\}$ be such that $b(t)$ is 1 if the server is busy at time t, and 0 otherwise, where $\mathcal{R}$ is the set of real numbers. The evolution of $x(t)$ is governed by a simple first-order model

$$\dot{x}(t) = \frac{b(t)-x(t)}{\tau}, x(0) = x_0 \tag{1}$$

where $\tau > 0$ is a time constant that determines the extent to which past utilization affects the current state of the server, and $x_0 \in [0,1]$ is the initial condition. The quantity $x(t)$ bounded by time interval $[t_1, t_2]$ denotes the utilization ratio of the server, i.e., the fraction of the recent history when the server was busy. Physically, $x(t)$ represents the perceived workload of the operator based on its recent utilization history within the time interval $[t_1, t_2]$. Equation (1) can be considered to be the continuum limit of the discrete time exponential moving window average by rewriting the time derivative in (1) from first principles

$$x(t + \Delta t) \approx \left(1 - \frac{\Delta t}{\tau}\right) x(t) + \frac{\Delta t}{\tau} b(t). \tag{2}$$

A simple moving window average model has been proposed in [16] for computing the utilization ratio. For other models of human mental workload, we refer the reader to [17]. The time constant τ corresponds to the inverse of the sensitivity of the operator to its recent utilization history: larger τ correspond to lower sensitivity and smaller τ correspond to higher sensitivity. Note that the set $[0,1]$ is invariant under the dynamics in (1) for any $\tau > 0$ and any $b: \mathcal{R} \to \{0,1\}$.

The service times bounded by time interval $[t_1, t_2]$ are related to the state $x(t)$ through a map $\mathcal{S}: [0,1] \to \mathcal{R}_{>0}$, where $\mathcal{R}_{>0}$ is the set of positive real numbers. If a task is allocated to the server at state x, then the amount of time required to perform unit work is given by $\mathcal{S}(x)$. Therefore, if the amount of work associated with a task allocated to the server at state x is w, then the service time on that task is $w\mathcal{S}(x)$. This linear decomposition of the total service time within time interval $[t_1, t_2]$ into the amount of work associated with the task and the rate of performing work with respect to the initial server state is an approximation to a more realistic scenario where the rate of performing work also depends on the amount of work such a model has been proposed in [4]. The linear decomposition that we use is reasonable especially for

small heterogeneity in the tasks. In our framework, the controller cannot interfere with the server while it is servicing a task. Hence, the only way in which the server state can be controlled is by scheduling the beginning of service of tasks after their arrival. Such controllers are called task-release controllers and will be formally characterized later on. In this paper, we assume that $\mathcal{S}(x)$ is positive valued, continuous, and convex. Let $\mathcal{S}_{min} := min\{\mathcal{S}(x) | x \in [0,1]\}$, and $\mathcal{S}_{max} := max\{\mathcal{S}(0), \mathcal{S}(1)\}$

The solution to (1) is $x(t) = e^{-t/\tau}(\int_0 (1/\tau) b(s) e^{s/\tau} ds + x_0)$. This implies that the server state $x(t)$ is increasing when the server is busy, i.e., when $b(t) = 1$, and decreasing when the server is not busy, i.e., when $b(t) = 0$. Note that $\mathcal{S}(x)$ is not necessarily monotonically increasing in x , since it has been noted in the human factors literature [18] that, for certain cognitive tasks demanding persistence, the performance [which in our case would correspond to the inverse of $\mathcal{S}(x)$] could increase with the state x when x is small. This is mainly because a certain minimum level of human behavior mental arousal is required for good performance. A well-known empirical law capturing such characteristics is the Yerkes-Dodson law [18] . A loose experimental justification of this server model in the context of human-in-the-loop systems is included in the related work [19], where $\mathcal{S}(x)$ for that setup was found to have a U-shaped profile. We will use that particular $\mathcal{S}(x)$ from [19] for various numerical illustrations in this paper. We provide further experimental evidence for this model in Section . It is important to note that the U-shaped relationship between the service time and the operator's utilization, as would be dictated, for example, by the Yerkes-Dodson law, falls within our assumptions on $\mathcal{S}(x)$ within time interval $[t_1, t_2]$ but it is not essential. In particular, our assumptions on $\mathcal{S}(x)$ also allow it to be monotonically increasing, decreasing, or even constant over $x \in [0,1]$.

3.2 Real Time Task Release Control Policy

We now describe task release control policies within the time interval for the Real Time Human Behavior queue. Without explicitly specifying its domain, a task release controller u acts like an ON-OFF switch at the entrance of the queue, e.g., see Fig. 1. In short, u is a task release control policy if $U(t) \in \{ON, OFF\}$ for all $t > 0$, and an outstanding task is assigned to the server if and only if the server is idle, i.e., when it is not servicing a task and $u = ON$. Let $\mathcal{U}$ be the set of all such task release control policies. For a given $\tau > 0$ and f_W , let $n_u(t, \tau, \lambda, f_W, x_0, n_0)$ be the queue length, i.e., the number of outstanding tasks, at time t, under task release control policy $u \in \mathcal{U}$, when the task arrival rate is λ and the server state and the queue length at time $t = 0$ are x_0 and n_0 , respectively. For brevity in notation, we will sometimes use the short hand notation $n_u(t)$ to denote the queue length at time t under task release control policy u when the other parameters are clear from the context. Note that we allow $\mathcal{U}$ to be quite general in the sense that it includes control policies that are functions of $\lambda, \mathcal{S}, x(t), f_W, \tau, n_u$, bounded by the time interval $[t_1, t_2]$ etc.

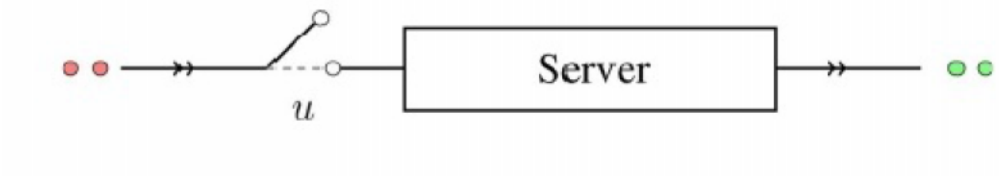

Figure 1: Real -Time Task Release Control Architecture

3.3 Define Human Behavior Stabilizable Arrival Rate

We now formally state the problem. Define the maximum stabilizable arrival rate within time interval $[t_1, +\infty]$ for policy u as

$$\lambda_{max}(\tau, F_W, u) :=$$

$$= sup\{\lambda : \lim_{t\to+\infty} sup n_u(t, \tau, \lambda, f_W, x_0, n_0)$$

$$< +\infty \forall x_0 \in [0,1], n_0 \in \mathcal{N} a.s.$$

The quantity $\lambda_{max}(\tau, f_W, u)$ will also be referred to as the throughput under policy u within time interval $[t_1, t_2]$. The maximum stabilizable arrival rate over all policies, or simply the real time throughput, is defined as $\lambda^*_{max}(\tau, f_W) := sup_{u\in\mathcal{U}} \lambda_{max}(\tau, f_W, u)$. For a given $\tau > 0$ and f_W , a task release control policy u is called maximally stabilizing if, for any $x_0 \in [0,1], n_0 \in \mathcal{N}$, $\lim sup_{t\to+\infty} n_u((t, \tau, \lambda, f_W, x_0, n_0) < +\infty$ for all $\lambda \leq \lambda^*_{max}(\tau, f_W)$ within time interval $[t_1, +\infty]$ almost surely. The primary objective in this paper is to compute the real time throughput and design a corresponding maximally stabilizing task release control policy for the dynamical queue whose server state evolves according to (1), and where $\mathcal{S}(x)$ is positive, continuous, and convex.

In this paper, we extensively focus on a specific class of task release control policies threshold policies. For a given $x^* \in [0,1]$, the x^* -threshold policy is defined as

$$u_{x^*}(t) = \begin{cases} ON, \\ if x(t) \leq x^* \\ OFF, \\ otherwise. \end{cases}$$

We prove that an appropriate threshold policy is maximally stabilizing when the tasks are homogeneous and utilize the threshold policies in the time interval to prove bounds on the real time throughput when the tasks are heterogeneous.

3.4 Simple Bounds on the Real Time Throughout

We start by deriving simple bounds on the real time throughput.

Proposition II.1: For any $\tau > 0$ and f_W , we have that $\lambda^*_{max}(\tau, f_W) \in [(\bar{w}\mathcal{S}(1))^{-1}, \bar{w}\mathcal{S}_{min})^{-1}]$

Proof: The time between the start of service of successive tasks consists of two parts: the time to actively service a task, and the time when the server is idle, as governed by the task release control policy. The upper bound on the throughput is obtained by neglecting the idle times and by assuming that the server spends the least amount of time to service every task. The lower bound is proven by considering the trivial policy $u(t) \equiv ON$ as follows. Assume, by contradiction, that the queue length grows unbounded under this policy for some initial condition for an arrival rate $(\bar{w}\mathcal{S}(1))^{-1} - \varepsilon$ for some $\varepsilon > 0$. For a queue length growing unbounded in the time interval $[t_1, +\infty]$, the server state exceeds $1 - \eta$ for any given $\eta > 0$ in some finite time $T[t_1, t_2]$. Note that the queue length remains bounded until $T[t_1, t_2]$. After $T[t_1, t_2]$, all the service times per unit work are upper bounded by $\mathcal{S}(1) + \theta$ where $\theta \geq 0$ depends on η through the continuity of $\mathcal{S}(x)$. One can select η and hence θ such that

$$(\bar{w}\mathcal{S}(1) + \bar{w}\theta)^{-1} > (\bar{w}\mathcal{S}(1))^{-1} - \varepsilon. \quad (3)$$

By the strong law of large numbers, with probability one, the average service time per task after T is upper bounded by $\bar{w}\mathcal{S}(1) + \bar{w}\theta$. Combining this with (3), we get that, after T, the arrival rate is strictly less than the mean service time with probability one and hence the queue length cannot grow unbounded with the time constraint $[t_1, t_2]$. This contradiction proves that the queue length remains bounded with probability one for an arrival rate $(\bar{w}\mathcal{S}(1))^{-1} - \varepsilon$ for any ε and for any initial condition, which in turn proves that $\lambda^*_{max}(\tau, f_W)$ is lower bounded by $(\bar{w}\mathcal{S}(1))^{-1}$.

The bounds with the time constraint $[t_1, t_2]$ obtained in Proposition II.1 can be shown to be tight for some simple cases. Consider first the case when $\mathcal{S} \equiv \varepsilon$ for some constant $\varepsilon > 0$. In this case, $\mathcal{S}(1) = \mathcal{S}_{min} = \varepsilon$ and hence Proposition II.1 implies that $\lambda^*_{max}(\tau, f_W) = (\bar{w}\varepsilon)^{-1}$ for all $\tau > 0$. Additionally, the trivial policy $u(t) \equiv ON$ is maximally stabilizing. Another simple case is when $\mathcal{S}(x)$ is nonincreasing. In this case, $\mathcal{S}(1) = \mathcal{S}_{min}$ and hence Proposition II.1 implies that $\lambda^*_{max}(\tau, f_W) = \bar{w}\mathcal{S}(1))^{-1}$ for all $\tau > 0$. One can show that the trivial policy $u(t) \equiv ON$ on is maximally stabilizing in this case as well. We now derive tighter bounds on the real time throughput and design corresponding maximally stabilizing task release control policies with time constraint .

3.5 Real Time Arriving Tasks

In this subsection, we consider the special case when the arriving tasks are homogeneous with time constraint , i.e., every task brings in exactly the same deterministic amount of work with it. Formally, we let $f_W(w) = \delta_{\bar{w}}(w)$ for some $\bar{w} \in [\mathcal{W}_1, \mathcal{W}_2]$. We start by studying specific types of equilibria that are associated with the trivial policy $u(t) \equiv ON$.

1)One Task Equilibria: Let $\boldsymbol{x_i}$ be the server state with the time $\boldsymbol{t_i}$ at the beginning of service of the $\boldsymbol{i}$ th task and let the queue length be zero at that instant. The server state upon the arrival of the $(\boldsymbol{i = 1})$ task is then obtained by integration of (1) over the time period $[\mathbf{0}, \mathbf{1}/\boldsymbol{\lambda}]$, with initial condition $\boldsymbol{x_0 = x_i}$. Let $\boldsymbol{x_i'}$ denote the server state when it has completed service of the $\boldsymbol{i}$ th task. Then, $\boldsymbol{x_i' = 1(1 - x_i)e^{-\bar{w}\mathcal{S}(x_i)/\tau}}$. Assuming that $\boldsymbol{\bar{w}\mathcal{S}(x_i) \leq 1/\lambda}$, we get that $\boldsymbol{x_{i+1} = x_i' e^{-(1/\lambda - \bar{w}\mathcal{S}(x_i))/\tau}}$, and finally $\boldsymbol{x_{i+1} = (1 - 1(1 - x_i)e^{-\bar{w}\mathcal{S}(x_i)/\tau})e^{(\bar{w}\mathcal{S}(x_i) - 1/\lambda)/\tau} = (x_i - 1 + e^{\bar{w}\mathcal{S}(x_i)/\tau}) \times e^{-(1/\lambda\tau)}}$. If $\boldsymbol{\lambda, \bar{w}}$ and $\boldsymbol{\tau}$ are such that $\boldsymbol{x_{i+1} = x_i}$, then under the trivial control policy $\boldsymbol{u(t) \equiv ON}$, the server state at the beginning of all the tasks after and including the $\boldsymbol{i}$ th task will be $\boldsymbol{x_i}$ and the queue length at most 1 with the time constraint $[\boldsymbol{t_0, t_i}]$. We then say that the server is at one-task equilibrium at $\boldsymbol{x_i}$. Therefore, for a given $\boldsymbol{\lambda, \bar{w}}$ and $\boldsymbol{\tau}$, the one-task equilibrium server states correspond to $\boldsymbol{x} \in [\mathbf{0}, \mathbf{1}]$ that satisfy $\boldsymbol{x = (x - 1 + e^{\bar{w}\mathcal{S}(x)/\tau})e^{-(1/\lambda\tau)}}$ and $\boldsymbol{\mathcal{S}(x) \leq (\bar{w}\lambda)^{-1}}$, i.e. $\boldsymbol{\mathcal{S}(x) = (\tau/\bar{w})\log(1 - (1 - e^{1/\lambda\tau})x)}$ and $\boldsymbol{\mathcal{S}(x) \leq (\bar{w}\lambda)^{-1}}$. Let us define a function $\boldsymbol{\mathcal{R}}$ as

$$\mathcal{R}(x, \tau, \bar{w}, \lambda) := \frac{\tau}{\bar{w}} \log\left(1 - (1 - e^{1/\lambda\tau})x\right). \quad (4)$$

For a given $\tau > 0$ and $\lambda > 0$, define the set of one-task equilibrium server states with the time constraint $[t_0, t_i]$ as

$$x_{eq}(\tau, \bar{w}, \lambda) := \{x \in [0,1] | \mathcal{S}(x) = \mathcal{R}(x, \tau, \bar{w}, \lambda)\}. \quad (5)$$

Note that we did not include the constraint $\mathcal{S}(x) \leq (\bar{w}\lambda)^{-1}$ in the definition of $x_{eq}(\tau, \bar{w}, \lambda)$ in (5). This is because this constraint can be shown to be redundant as follows. Equation (4) implies that, for any $\tau > 0$ and $\lambda > 0$, $\mathcal{R}(x, \tau, \bar{w}, \lambda)$ is strictly increasing in x and hence $\mathcal{R}(x, \tau, \bar{w}, \lambda) \leq \mathcal{R}(1, \tau, \bar{w}, \lambda) = (\lambda\bar{w})^{-1}$ for all $x \in [0,1]$. Therefore, $\mathcal{S}(x_{eq}(\tau, \bar{w}, \lambda)) = \mathcal{R}(x_{eq}(\tau, \bar{w}, \lambda), \tau, \bar{w}, \lambda) \leq (\lambda\bar{w})^{-1}$.

The strict convexity of $\mathcal{S}(x) - \mathcal{R}(x, \tau, \bar{w}, \lambda)$ in x, which follows from the convexity assumption on $\mathcal{S}(x)$ and the strict concavity of $\mathcal{R}$ in x from (4) within time interval $[t_1, t_i]$, implies that the cardinality of $x_{eq}(\tau, \bar{w}, \lambda)$ can take on values 0, 1, and 2. For a given $\tau > 0, \bar{w} > 0$, and $\lambda > 0$, let $x_{eq,1}(\tau, \bar{w}, \lambda)$ be the smaller element of $x_{eq}(\tau, \bar{w}, \lambda)$ if it is not empty and let $x_{eq,2}(\tau, \bar{w}, \lambda)$ be the other element if the cardinality of $x_{eq}(\tau, \bar{w}, \lambda)$ is 2. One can show that $x_{eq,1}(\tau, \bar{w})$, if it exists, is a stable equilibrium point and $x_{eq,2}(\tau, \bar{w})$, if it exists, is an unstable equilibrium point. Formally, one can show that, if $x_{eq,1}(\tau, \bar{w})$ and $x_{eq,2}(\tau, \bar{w})$ exist, then we have the following.

1) For any $\tau > 0$ and $\bar{w} > 0$, the set $(x_{eq,2}(\tau, \bar{w}), 1]$ is invariant and is not in the region of attraction of $x_{eq,1}(\tau, \bar{w})$ or $x_{eq,2}(\tau, \bar{w})$.

2) There exists a $\tau' > 0$ such that for all $\tau > \tau'$, the set $[0, x_{eq,2}(\tau, \bar{w}))$ is invariant for all $\tau > \tau'$. Moreover, in the limit as $\tau \to +\infty$, the set $[0, x_{eq,2}(\tau, \bar{w}))$ is the region of attraction of $x_{eq,1}(\tau, \bar{w})$.

We introduce a couple of additional definitions. For a given $\tau > 0$ and $\bar{w} > 0$, let

$$\begin{aligned} &\lambda_{eq}^{max}(\tau, \bar{w}) \\ &max\{\lambda > 0 | x_{eq}(\tau, \bar{w}, \lambda) \neq \emptyset\} \\ &x_{th}(\tau, \bar{w}) \\ &x_{eq,1}\left(\tau, \bar{w}, \lambda_{eq}^{max}(\tau, \bar{w})\right). \end{aligned} \tag{6}$$

We now argue that the definitions in (6) with the time constraint $[t_1, t_2]$ are well posed. Consider the function $\mathcal{S}(x) - \mathcal{R}(x, \tau, \bar{w}, \lambda)$. Since $\mathcal{R}(0, \tau, \bar{w}, \lambda) = 0$ for any $\tau > 0, \bar{w} > 0$, and $\lambda > 0$, and $\mathcal{S}(0) > 0$, we have that $\mathcal{S}(0) - \mathcal{R}(0, \tau, \bar{w}, \lambda) > 0$ for any $\tau > 0, \bar{w} > 0$, and $\lambda > 0$. Since $\mathcal{R}(1, \tau, \bar{w}, \lambda) = (\bar{w}\lambda)^{-1}$, $\mathcal{S}(1) - \mathcal{R}(1, \tau, \bar{w}, \lambda) < 0$ for all $\lambda < (\bar{w}\mathcal{S}_{max})^{-1}$. Therefore, by the continuity of $\mathcal{S}(x) - \mathcal{R}(x, \tau, \bar{w}, \lambda)$, the set of equilibrium server states, as defined in (5), is nonempty for all $\lambda < (\bar{w}\mathcal{S}_{max})^{-1}$. Moreover, since $\mathcal{R}(x, \tau, \bar{w}, \lambda) \leq \mathcal{R}(1, \tau, \bar{w}, \lambda) = (\bar{w}\lambda)^{-1}$ for all $x \in [0,1]$, $\mathcal{S}(x) - \mathcal{R}(x, \tau, \bar{w}, \lambda) \geq (\bar{w}\mathcal{S}_{min})^{-1}$ for all $x \in [0,1]$. Therefore, for all $\lambda > (\bar{w}\mathcal{S}_{min})^{-1}$, the set of equilibrium states, as defined in (5), is empty. Hence, $\lambda_{eq}^{max}(\tau, \bar{w})$ and $x_{th}(\tau, \bar{w})$ are well defined.

In the rest of the paper, we will restrict our attention to those $\tau, \bar{w} > 0$, and $\mathcal{S}(x)$ for which $x_{th}(\tau, \bar{w}) < 1$. Loosely speaking, this is satisfied when $\mathcal{S}(x)$ is increasing on some interval in $[0,1]$ and the increasing part is steep enough. It is reasonable to expect this assumption to be satisfied in the context of human operators with time constraint whose performance deteriorates quickly at very high utilizations. The implications of the case when $x_{th}(\tau, \bar{w}) = 1$ are discussed briefly at appropriate places in the paper.

2)Lower Bound on the Real Time Throughput: We start by analyzing the real time throughput under a specific task release control policy. In particular, we consider the $\boldsymbol{x_{th}(\tau, \bar{w})}$ threshold

policy, where $x_{th}(\tau, \bar{w})$ is as defined in (6).

Theorem III.1: For any $\tau > 0, \bar{w} > 0, x_0 \in [0,1], n_0 \in \mathcal{N}$, and $\lambda \leq \lambda_{eq}^{max}(\tau, \bar{w})$, if $x_{th}(\tau, \bar{w}) < 1$, then we have that $\limsup_{t\to+\infty} n_u(t, \tau, \lambda, \delta_{\bar{w}}, x_0, n_0) < +\infty$ with u being the $x_{th}(\tau, \bar{w})$ threshold policy. The proof of this result, which can be found in [20] .

3) Upper Bound on the Real Time Throughput: We now prove that the $x_{th}(\tau, \bar{w})$ threshold policy with time constraint $[t_1, t_{th}]$ is indeed maximally stabilizing by showing that no other task release control policy in $\mathcal{U}$ gives more real time throughput. Recall that a task release control policy u is maximally stabilizing within time interval $[t_0, t_\infty]$ in this setup if, for any $x_0 \in [0,1], n_0 \in \mathcal{N}$, $\limsup_{t\to+\infty} n_u(t, \tau, \lambda, \delta_{\bar{w}}, x_0, n_0) < +\infty$ for all $\sup_{u\in\mathcal{U}} \lambda_{max}(\tau, \delta_{\bar{w}}, u)$, where $\lambda_{max}(\tau, \delta_{\bar{w}}, u)$ is the throughput under policy u. We emphasize here that the allowable set of control policies $\mathcal{U}$ is pretty general and in particular, it includes, but is not limited to, threshold policies.

4. Conclusion

In this paper, we presented a real time human behavior queue framework as a formal approach to task management for human operators. Inspired by empirical laws, we considered a novel human behavior queue model for human operators, where the service times are dependent on the state of a simple underlying real time system. We studied the stability of such human behavior queues under deterministic interarrival times and real times. For homogeneous tasks, we proved that a task release control policy that releases a task to the server only when its state is below an appropriately chosen threshold value gives the maximum throughput. For heterogeneous tasks, we showed that the throughput strictly increases with the introduction of heterogeneity. The deterministic interarrival time assumption in our analysis is not binding and the results extend to the case where the interarrival times are sampled identically and independently from a common distribution having bounded variance. We also reported preliminary empirical evidence to justify the real time human behavior queue model for human operators.

We have extended the related work based on human behavior queuing system for accurate modeling of wireless IP traffic behavior through presenting a novel scheduling scheme called as Best Scheduling Algorithm (BSA). The simulation results clearly indicate that our proposed scheduling algorithm outperforms the traditional scheduling schemes such as priority and round-robin. The BSA provides a preferential treatment to real time applications by offering a very low delay but at the same time, this preference is not up to that extent that generic data applications are starving for bandwidth. In our future work, we are intending to explore the possibility of practical implementation of proposed BSA in different 4G wireless networks.

References

[1] G. Armitage, "Quality of Service in IP Networks", MTP, pps 105-138, 2004.

[2] K. Savla, T. Temple, and E. Frazzoli, Human-in-the-loop vehicle routing policies for dynamic environments, Proceeding. IEEE Conf. Decision Control, pp. 1145¡V1150, 2008.

[3] J. H. Dshalalow, Ed., BQueueing systems with state dependent parameters,[in Frontiers in Queuing Models and Applications in Science and Engineering. Boca Raton, FL: CRC Press, 1997.

[4] R. Bekker and S. C. Borst, BOptimal admission control in queues with workload-dependent service rates,[Probab. Eng. Inf. Sci., vol. 20, pp. 543¡V570, 2006.

[5] B. Tsybakov and N. D. Georganas, "Self-Similar traffic and upper bounds to buffer overflow in ATM queue", Performance Evaluation, 36, pps. 57-80, 1998.

[6] R. Addie, M. Zukerman and T. Naeme, "Fractal traffic: measurements, modeling and performance evaluation", Proceeding IEEE INFOCOMM 95, pp. 977-984, 1995.

[7] M. Zukerman et al, "Analytical Performance Evaluation of a Two Class DiftServ Link", IEEE ICS, vol. 1, pp. 373-377, 25-28 Nov. 2002.

[8] J. Zhang, "Performance study of Markov modulated fluid flow models with priority traffic", in Proc. IEEE INFOCOM 93, San Francisco, CA, pps 10-17, Mar. 30-Apr. 1, 1993.

[9] Do. Young. Eun and Ness. B. Shroff, "A measurementanalytical approach for QoS estimation in a network based on dominant time scale" in IEEE/ACM Trans. on Networking, vol. II, No. 2, pps. 222-235, April 2003.

[10] A. I. Elwalid and D. Mitra, "Fluid Models for analysis and design of statistical multiplexing with loss priorities on multiple classes of bursty traffic" in Proc. IEEE INFOCOM 92, Florence, Italy, pps 415-425, 1992.

[11] G. L. Choudhury, K. K. Leung and W. Whitt, "An inversion algorithm to compute blocking probabilities in loss networks with state-dependant rates", IEEE/ACM Transactions on Networking, vol. 3. pp. 585-601, 1995.

[12] C. F. Chou et ai, "Low Latency and efficient packet scheduling for streaming applications" IEEE International Conference on Communications, Vol. 4, pp. 1963-1967, 20-24 June, 2004.

[13] J. M. Chung, H. M. Soo, "Analysis of Non Preemptive Priority Queueing of MPLS networks with Bulk Arrivals", IEEE MWSCAS, vol. 3. pps 81-84, 4-7 Aug. 2002.

[14] M. Iftikhar et aI, “SLAs parameter negotiation between heterogeneous 4G wireless network operators”, Elseveir Journal of Pervasive and Mobile Computing, vol. 7, issue 5, pps. 525-544, October 2011.

[15] M. Iftikhar et aI, “Towards the formation of comprehensive SLAs between heterogeneous wireless DiffServ domains”, Springer Journal of Telecommunication Systems, 42: 179-199, 2009.

[16] M. L. Cummings and C. E. Nehme, BModeling the impact of workload in network centric supervisory control settings, Proceeding. 2nd Annu. Sustaining Performance Under Stress Symp., College Park, MD, Feb. 2009.

[17] P. A. Hancock and N. Meshkati, Eds., Human Mental Workload, vol. 52, Advances in Psychology. Amsterdam, The Netherlands: Elsevier Science, 1988.

[18] R. M. Yerkes and J. D. Dodson, BThe relation of strength of stimulus to rapidity of habit-formation, J. Comparative Neurol. Psychol., vol. 18, pp. 459¡V482, 1908.

[19] K. Savla, C. Nehme, T. Temple, and E. Frazzoli, BEfficient routing of multiple vehicles for human-supervised services in a dynamic environment, Proceeding AIAA Conf. Guid. Navig. Control, Honolulu, HI, 2008, Paper AIAA 2008-6841.

[20] K. Savla and E. Frazzoli, Maximally stabilizing admission control policy for a dynamical queue, IEEE Trans. Autom. Control, vol. 55, no. 11, pp. 2655¡V2660, Nov. 2010.

Authors

Tony Tsang received the BEng degree in Electronics & Electrical Engineering with First Class Honours in U.K., in 1992. He received the Ph.D from the La Trobe University (Australia) in 2000. He was awarded the La Trobe University Postgraduation Scholarship in 1998. He is a Lecturer at the Hong Kong Polytechnic University. Prior to joining the Hong Kong Polytechnic University, Dr. Tsang earned several years of teaching and researching experience in the Department of Computer Science and Computer Engineering, La Trobe University. He has numerous publications in international journals and conferences and is a technical reviewer for several international journals and conferences. His research interests include mobile computing, networking, protocol engineering and formal methods. Dr. Tsang is a member of the ACM and the IEEE.