Fast Queuing Policies for Multimedia Applications

Duong Nguyen-Huu
School of EECS
Oregon State University
Corvallis, OR 97331
Email: nguyendu@eecs.oregonstate.edu

Thai Duong
School of EECS
Oregon State University
Corvallis, OR 97331
Email: duong@eecs.oregonstate.edu

Thinh Nguyen
School of EECS
Oregon State University
Corvallis, OR 97331
Email: thinhq@eecs.oregonstate.edu

***Abstract*—We present an analytical framework for providing Quality of Service (QoS) using queuing policies that achieves a given target distribution of packets in a network queue. To a large extent, the stationary distribution of packets in the queue resulted from employing a certain queuing policy directly controls the typical QoS metrics for multimedia applications. Therefore, using the packet distribution in the queue as the metric, the proposed framework allows for a more general and precise control of QoS beyond the standard metrics such as bandwidth, jitter, loss, and delay. Moreover, the proposed framework aims to find a fast queuing policy that achieves a given target stationary distribution. This fast adaptation is especially useful for multimedia applications in fast-changing network conditions. As an example, we present a general procedure for obtaining a queuing policy that optimizes for a given arbitrary objective along with the standard QoS requirements. Both theory and simulation results are presented to verify our framework.**

***Index Terms*—QoS, Queuing theory, Distribution Shaping, Convex Optimization**

I. Introduction

Since the development of packet-switched networks in early 1960s, queuing theory has been a critical part in the performance analysis for most if not all the modern transmission protocols. The performance of TCP/IP protocols for example, can be analyzed in the language of queuing theory [1]. Many current wireless transmission protocols such as the IEEE 802.11 protocols owed their analysis to queuing theory. In fact, queues are so universal that they are virtually found in every communication devices from the core Internet routers and broadband modems to wireless LAN and cellular devices. It is therefore, not a surprise that a point-to-point data flow is typically modeled as a single queue or a network of queues. Understanding the dynamics of packets in queues over time as a result of employing certain queuing policy, enables the system engineers to characterize and to predict various properties of the data flow such as bandwidth, packet loss and delay. With the advent of multimedia communication applications that require certain levels of Quality of Service (QoS), e.g., requirements on minimum bandwidth, maximum jitter, delay, or loss, the role of queuing policy is becoming increasingly more important.

Queue and Queuing Policy: In a typical packet-switched network, the instantaneous arrival rates of packets at an intermediate router can vary significantly, and a packet loss occurs when the arrival rate exceeds the sending rate at a router. Therefore, a queue or a buffer is used to temporarily store a burst of incoming packets in an attempt to prevent packet loss. These packets waits for their turns in the queue to be transmitted to the next hop, or to read by an application if the queue is located at a receiving end device. Queuing policy is a mechanism used to control various operations of a queue that govern the packet's entrance, departure, and drop. It is directly responsible for shaping the dynamics of packets in the queue which characterizes the delay, loss, and bandwidth of a flow. Depending on certain constraints, some queuing polices are more limited in their operations than others. For example, a simple queuing policy is the First In First Out (FIFO) scheme which is typically implemented at the Internet core routers. A router using FIFO policy sends out packets in the order of their arrivals as fast as possible. Packets arriving at the router are dropped when the queue is full. One important observation is that the FIFO scheme has no ability to control the sending or dequeue rate, nor it has the ability to provide feedback to the upstream node for adjusting the incoming or enqueue rate. On the other hand, a more sophisticated queuing policy would be able to control, at least probabilistically, the dequeue rate and the enqueue rate possibly via feedback in order to achieve some given objectives such as queue stability or average queue length. For example, the well-known end-to-end flow control of TCP use feedback (ACK message) to control the sending (enqueue) rate. The IEEE 802.11 protocol family also employs feedback in the form of collisions to adjust the sending rates appropriately.

Queuing policies are most prevalent to achieve QoS for many applications sharing the same resources. Typically, arrival packets from different applications are enqueued in separate virtual queues. A queuing policy is then determine the next packet from which queue to be processed based on the queue lengths, QoS requirements of different applications. For example, fair queueing [2] and weighted queuing [3] are two well-known queuing policies to achieve throughput stability and given QoS application requirements.

Beyond network protocols, queues are also extensively used in rate control for video coding [4] [5]. The objective of rate control is to produce a coded video bit stream with a certain average bit rate and variance. In this setting, a ”conceptual”' queue is connected to a video encoder. The feedback from the queue to the video encoder is used by the video encoder to adjust the coded video bit rate using the coding parameters such as quantization level and coding mode appropriately.

978-1-4673-5775-3/13/$31.00

In this paper, we consider a general class of queuing polices that allows for the ability to adjust the sending and receiving rates *probabilistically*. The probabilistic framework arises naturally from the unavoidable uncertainties in when and how fast packets arrive due to the fluctuations in network traffic. Furthermore, in some scenarios the ability to send packets out (de-queue) successfully at any time is probabilistic. For example, in a Wi-Fi network, a wireless node might not be able to successfully send out a packet (de-queue) at a certain time slot due to possible collision with other node's transmission [6]. Also, its random back-off mechanism after a collision can in fact be viewed as a dequeuing operation with a certain probability. We note that almost of the current queuing policies whether they are deterministic or probabilistic, or have ability to control the receiving rates or not, are subset of the general class of queuing policies that we consider. In this paper, we also limit our discussion to the analysis of queuing policy for a single queue. We believe the analysis for this simple case is still useful since it is applicable to providing QoS in the last mile scenario or single-hop networks such as a Wi-Fi or access networks.

Our contributions include an analytic framework for providing Quality of Service (QoS) using a *fast* queuing policy that achieves a given target distribution of packets in a network queue. Using the packet distribution in the queue as the metric, the proposed framework allows for a more general and precise control of QoS beyond the standard metrics such as bandwidth, jitter, loss, and delay. The fast adaptive queuing policies are especially useful for multimedia applications in fast-changing network conditions. Finally, we show how an even faster queuing policy can be achieved when the queuing policy only needs to produce the stationary distribution that is ϵ-close to the given target stationary distribution. Our framework is developed based on the theory of fast mixing chain and convex optimization. As an example, we present a general procedure for obtaining a queuing policy that optimizes for a given arbitrary objective along with the standard QoS requirements.

Our paper is organized as follows. In Section II, we provide some background on the Markov Chain and queuing theories as they are necessary for the development of our proposed framework. In Section III, we describe a convex optimization framework with multiple formulations for finding fast queuing policies. In Section IV, we show an application of our framework to finding a queuing policy that optimizes for a given objective for a flow while ensuring the mean and variance of queuing delay are within given bounds. Finally, we provide a few concluding remarks.

II. Preliminaries

A. *Queues, Markov Chain, and Mixing Times*

A finite and discrete Markov chain is a set of sequence of random variables X_1, X_2, $\ldots X_n$ such that given the present states, the past and the future states are independent. A finite state time-homogeneous Markov chain is characterized by a time-invariant transition probability matrix P. In the context of queue, let N be the number of the maximum physical queue length, X_n be the number of packets in the queue at time step n, then dynamics of the number of packets in the queue can be mathematically represented by a Markov chain with a square tridiagonal probability matrix $P^{N\times N}$.

In order to quantify "fast" queuing policy, i.e., how fast a queuing policy drive an initial distribution to a given target stationary distribution, it is necessary to define a similarity measure between two distributions. One common similarity measure is the total variance distance defined below:

Definition 1 (Total variation distance): For any two probability distributions ν and π on a finite state space Ω, we define the total variation distance as:

$$\|\nu-\pi\|_{TV}=\frac{1}{2}\sum_{i\in\Omega}|\nu(i)-\pi(i)|\,.$$

We now use the similarity measure to define an important notion called mixing time below:

Definition 2 (Mixing time): For a discrete, aperiodic and irreducible Markov chain with transition probability P and stationary distribution π, given an $\epsilon>0$, the mixing time $t_{mix}(\epsilon)$ is defined as

$$t_{mix}(\epsilon) = \inf\left\{n:\|\nu^TP^n-\pi^T\|_{TV}\le\epsilon, \text{for all probability distributions } \nu\right\}.$$

Essentially, the mixing time of a discrete time Markov chain is the minimum number of time step n until the total variance distance between the n-step distribution and the stationary distribution is less than ϵ. We will use the mixing time to characterize the convergence rate of a queuing policy. One of the successful techniques for bounding the mixing time of a stochastic matrix is via its spectral characterization, i.e., its eigenvalues.

Eigenvalues and Eigenvectors. Denote the set of eigenvalues of a stochastic matrix P in non-increasing order:

$$1=\lambda_1(P)\ge\lambda_2(P)\ge\cdots\ge\lambda_{|\Omega|}(P)\ge-1$$

Definition 3 (Second largest eigenvalue modulus): The second largest eigenvalue modulus (SLEM) of a matrix P is defined as:

$$\mu(P)=\max_{i=2,\ldots,|\Omega|}|\lambda_i(P)|=\max\{\lambda_2(P),-\lambda_{|\Omega|}(P)\} \quad (1)$$

In this paper, we also make use the reversibility property of Markov chain defined as follows:

Definition 4 (Reversible Markov Chain): A discrete Markov chain with a transition probability P is said to be reversible if

$$P_{ij}\pi(i)=P_{ji}\pi(j) \quad (2)$$

We now show an important bound that relates the mixing time of the Markov chain to the SLEM of a reversible matrix P.

Theorem 1 (Bound on mixing time): [7]. Let P be the transition matrix of a reversible, irreducible and aperiodic Markov chain with state space Ω, and let $\pi_{min}:=\min_{x\in\Omega}\pi(x)$. Then

$$t_{mix}(\epsilon)\le\frac{1}{1-\mu(P)}\log\big(\frac{1}{\epsilon\pi_{min}}\big). \quad (3)$$

It is not difficult to see that from Theorem 1, the error ϵ reduces over time at a rate of no greater than $\frac{e^{-(1-\mu(P))t}}{\pi_{min}}$. Thus, finding the matrix P with minimum $\mu(P)$ would result in the fastest convergence time. Next, we discuss previous results on how to find reversible matrices or queuing policies with fast convergence rates.

B. Finding Queuing Policy with Fast Convergence Rate

For a reversible, irreducible, aperiodic chain P with stationary distribution π, one can show that

$$\mu(P) = ||D_{\pi}^{1/2} P D_{\pi}^{-1/2} - \sqrt{\pi}(\sqrt{\pi})^T||_2, \tag{4}$$

where D_π denotes the square diagonal matrix whose diagonal entries are taken from each elements of π, and $||.||$ denote l_2-induced matrix norm [8][9].

Then given a target distribution π^*, it is not difficult to see that $\mu(P)$ is a convex function in P. Thus, the problem of finding the reversible matrix P with the smallest SLEM, or Fastest Mixing Markov Chain (FMMC) is the following convex optimization:

FMMC framework.

$$\begin{aligned} &\text{Minimize } ||D_{\pi^*}^{1/2} P D_{\pi^*}^{-1/2} - \sqrt{\pi^*}(\sqrt{\pi^*})^T||_2 \\ &\text{Subject to : } \begin{cases} P\mathbf{1} = \mathbf{1} \\ D_{\pi^*} P = P^T D_{\pi^*} \end{cases} \end{aligned} \tag{5}$$

We note that the first constraint guarantees the matrix P to be a valid transition probability matrix, while reversibility is enforced in the second constraint. In the framework, P is the only optimization variable.

We also considered the an extension of the FMMC problem called the EFMMC problem. In the EFMMC problem the goal is to produce even a faster mixing Markov chain than the one obtained by the FMMC. However, the resulted stationary distribution is no longer exactly the given target distribution, but is an ϵ-approximation to the target stationary distribution. Specifically, we have shown that the solution of EFMMC can be obtained using the following convex optimization:

Extended FMMC framework.

$$\begin{aligned} &\text{Minimize } ||D_{\pi^*}^{1/2} P D_{\pi^*}^{-1/2} - \sqrt{\pi^*}(\sqrt{\pi^*})^T||_2 \\ &\text{Subject to : } \begin{cases} P\mathbf{1} = \mathbf{1} \\ ||\pi^{*T} P - \pi^{*T}||_2 \leq \delta \\ \text{Other convex constraints on } P. \end{cases} \end{aligned} \tag{6}$$

In our previous work [9], we showed that by choosing appropriate appropriate value of $\delta(\epsilon)$, we can guarantee the solution to our EFMMC problem will produce a stationary distribution π that is ϵ- approximation of π^*, i.e., $|\pi - \pi^*| \leq \epsilon$.

Now we make the connection to the queuing policy and reversible matrix with the following Proposition:

Proposition 1: Any tridiagonal transition matrix corresponds to a reversible Markov Chain.

Since every queuing policy corresponds to a tridiagonal transition probability matrix, from the Proposition 1 all the queuing policies that we considered are reversible. Also, it is not difficult to add in additional convex constraints to ensure that the solutions of the convex optimization problems above to have the solution as a tridiagonal matrix.

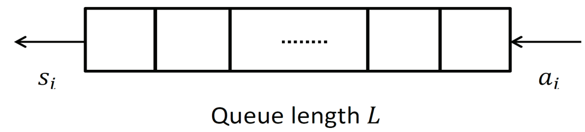

Figure 1: Discrete queue model

However, it is important to note that for a given tridiagonal transition probability matrix, there might not be a valid queuing policy for specific settings. Therefore, even if one finds the fastest reversible matrix for either two convex formulations above, there might not be readily a feasible queuing policy. In the next section, we show a method for finding an approximate queuing policy based on the P found in the two formulations.

III. TRIDIAGONAL MATRICES AND QUEUE POLICY

Depending on specific settings, the tridiagonal will not produce a valid queuing policy, more precisely, produce a feasible way for controlling the enqueue and dequeue rates. Let us consider the following scenario in which the arrival and departure rates at the queue can be controlled to some extent by a queuing policy. Let us assume that as a result of a queuing policy, the probabilities of a packet arriving at the queue and departing from the queue when the queue length is i, are a_i and s_i, respectively. We assume that packets can only arrive and depart at the beginning of each discrete time slot. We note that the ability to control the arrival rate seems impossible for physical queues in the Internet routers, however, it is frequently implemented in high level network protocols such as TCP in which virtual queues are typically used to provide feedback to the sender for the purpose of rate control. Importantly, we note that the proposed class of queuing policies also covers the *open loop* case where there the queuing policy cannot control the arrival rate. Using this queuing model as shown in Fig. 1, let us denote:

- $|\Omega|$: Maximum queue length
- $s = (s_0, \ldots, s_{|\Omega|})$ where $s_0 = 0$: Departing probability vector
- $a = (a_0, \ldots, a_{|\Omega|})$ where $a_{|\Omega|} = 0$: Arrival probability vector.

Then it is not difficult to see that the dynamics of the number of packets in a queue over time is governed by a discrete Markov chain with the transition probability matrix below:

$$Q = \begin{pmatrix} 1-a_0 & a_0 & & \\ s_1(1-a_1) & 1-s_1-a_1+2s_1a_1 & (1-s_1)a_1 & \\ & \ddots & \ddots & \ddots \\ & & s_{|\Omega|} & 1-s_{|\Omega|} \end{pmatrix} \tag{7}$$

Note that for each non-zero entry of each row, the left, middle, and right entries denote the probabilities that the number of packets in the queue decreases by 1, stays the same, or increases by 1, respectively.

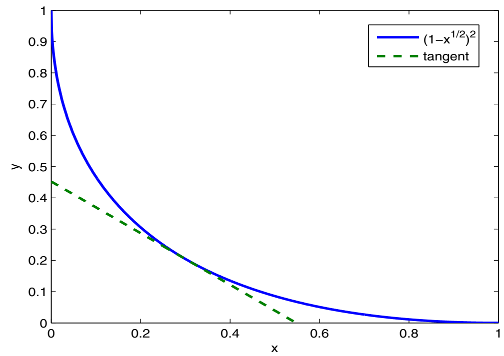

Figure 2: Tangent at $x_0 = 0.3$ of the function $f(x) = (1 - \sqrt{x})^2$

Now, let us compare the above matrix Q to the matrix P which is the solution obtained from the problem FMMC (or EFMMC) above. In general, P is a tridiagonal matrix with the entries: r_i, q_i, p_i.

$$P = \begin{pmatrix} r_0 & p_0 & & & \\ q_1 & r_1 & p_1 & & \\ & \ddots & \ddots & \ddots & \\ & & q_{|\Omega|} & r_{|\Omega|} & \end{pmatrix} \quad (8)$$

The main challenge is how to find the corresponding s_i and a_i, i.e., enqueue and dequeue rates for given r_i, q_i, p_i. It turns out that s_i and a_i might be negative or complex numbers which cannot be used in a feasible queuing policy. However, we can determine the conditions on q_i and p_i for which there exist real and non-negative solutions for s_i and a_i, leading to a feasible queuing policy. We proceed to derive the conditions as follows.

From (7) and (8), we need to solve these following equations:

$$\begin{cases} s_i(1-a_i) = q_i \rightarrow a_i = 1 - q_i/s_i \\ (1-s_i)a_i = p_i \rightarrow a_i = p_i/(1-s_i) \end{cases}$$

$$\iff 1 - q_i/s_i = p_i/1 - s_i \text{ for } i = 1, \ldots, |\Omega| - 1$$

$$\iff (1-s_i)s_i = (1-s_i)q_i + s_i p_i \text{ for } i = 1, \ldots, |\Omega| - 1$$

$$\iff s_i^2 - s_i(1+q_i-p_i) + q_i = 0 \text{ for } i = 1, \ldots, |\Omega| - 1 \quad (9)$$

In order to guarantee the existence of feasible solution of (9), we need:

$$\Delta = (1+q_i-p_i)^2 - 4q_i \geq 0 \text{ for } i = 1, \ldots, |\Omega| - 1 \quad (10)$$

It appears that we can add these constraints directly to the two convex formulations above. However, these constraints are not convex, thus making it hard to solve in general. Therefore, our approach is to relax (10) by making it a convex constraint as follows.

$$(1+q_i-p_i)^2 - 4q_i \geq 0 \iff 1+q_i-p_i > 2\sqrt{q_i} \text{ since } q_i > 0$$

$$\iff (1-\sqrt{q_i})^2 > p_i \quad (11)$$

Consider function $f = (1-\sqrt{x})^2$ for $x \in (0,1)$, we can find an approximate lower bound function $f(.)$ in the form of tangent $y = a(x_0)x + b(x_0)$ where $a = f'(x_0)$ for $x_0 \in (0,1)$ and $f'(x) = (\sqrt{x}-1)/\sqrt{x}$ (See (Fig. 2)).

Hence, (11) is equivalent to the following convex constraints:

$$a(x_0)q_i + b(x_0) > p_i \text{ for } i = 1, \ldots, |\Omega| - 1 \quad (12)$$

Now, we can incorporate these constraints in (12) to the FMMC and/or EFMMC problems, and still have convex formulations to find feasible queuing policies.

IV. Optimizing a Given Objective via Queuing Policy

A. Approach Illustration

In this section, we provide an example of applying our proposed framework to find fast queuing policy that optimizes a given objective while still satisfying other standard QoS requirements. Our approach consists of two steps. In the first step, we find a stationary distribution π^* that optimizes a given objective subject to all the given constraints assuming that the given objective and the constraints are convex in π, and thus π^* can be determined efficiently. In the second step, we substitute π^* into either the FMMC or EFMMC with the convex constraints in (12) to find the fastest queuing policy. We give a specific example below.

Step 1. Let X be discrete random variable representing the number of packets in the queue ($X \in [0, \ldots, L]$) and the number of states $|\Omega| = L + 1$.
Suppose a video application requires that the queuing delay average and second moment must be bounded within a range. For example,

$$\begin{cases} E[X] < Y1 \\ E[X^2] < Y2 \end{cases}$$

Then $E[X]$ and $E[X^2]$ can be computed from the stationary distribution π:

$$\begin{cases} E[X] = \sum_{x=0}^{L} \pi(x)x \\ E[X^2] = \sum_{x=0}^{L} \pi(x)x^2 \end{cases}$$

Furthermore, suppose that there is a cost function $c(x)$ where x denotes the number of packets in the queue. $c(x)$ could be any arbitrary function that might represents energy, resources that depend on the queue occupancy. Now, suppose we want to minimize the total expected cost,

$$T = \sum_{x=0}^{x=L} c(x)\pi(x).$$

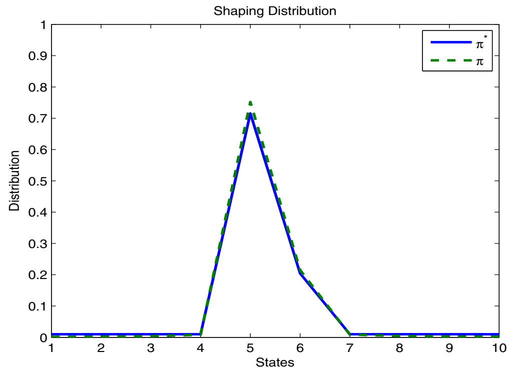

Figure 3: Target and resulted distribution

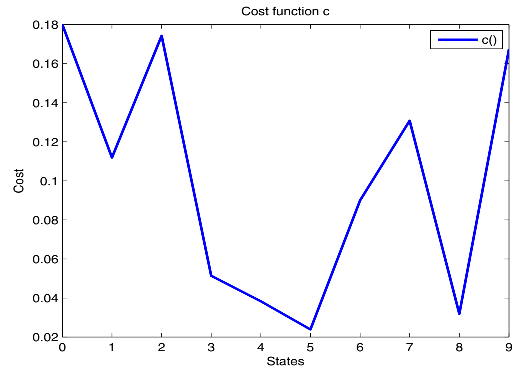

Figure 4: Cost function $c(x)$

Then the optimization problem can be formulated as follows.

$$\text{Minimize} \sum_{x=0}^{x=L} c(x)\pi(x)$$
$$\text{Subject to} : \begin{cases} \sum_{x=0}^{L} \pi(x)x < Y1 \\ \sum_{x=0}^{L} \pi(x)x^2 < Y2 \\ \sum_{x=0}^{L} \pi(x) = 1 \\ \pi_{min} < \pi(x) \ \forall x = 0, 1, \ldots, L \end{cases} \quad (13)$$

Step 2. The solution of (13) gives us the target stationary distribution π^* satisfying the QoS requirements and the given objective. Now, we apply the FMMC and EFMMC formulations to find tridiagonal matrices P with fast mixing rates. Next, using P and the method shown in Section III, we can find the matrix Q, i.e., the dequeuing and dequeuing rates as a function of the number of packets in the queue. This will result in a queuing policy that achieves the target distribution quickly, also satisfies the QoS requirements.

B. Performance Evaluation

In this section, we present the performance evaluations of our approach using the example above with specific parameters. In practice, the specific cost function $c(x)$ is application and device dependent. For example, $c(x)$ could model the power consumption of a smart phone which might be inversely quadratic in the number of packets (x) in its queue. In other scenarios, on average a device might consume more power if more packets are waiting in the queue due to some circuitry implementation. On the other hand, implementing a fast sending rate to result in fewer packets in the queue might also leads to more power consumption, perhaps due to faster clock rate. Thus, there might be some sweeping operating range in order to minimize the consumption power. Rather than choosing a specific cost function $c(x)$, to demonstrate the flexibility of our approach, the cost function $c(x)$ is chosen arbitrarily as shown in Fig. 4. The first and second moments are bounded by:

$$Y1 = 5; Y2 = 19.$$

To guarantee aperiodic and irreducible property, we require the minimum value of π to be a small non-zero value. Specifically, we set:

$$\pi_{min} = 0.01.$$

The arrival rate vector and the service rate vector are assumed to be bounded by $(0.01, 0.09)$. We assume the maximum physical queue length $L = 9$ or the number of states $|\Omega| = 10$.

We also note that our framework does not assume the convex property of $c(x)$ since regardless of $c(x)$, our objective is always convex in $\pi(x)$. Thus, the network designers can arbitrarily choose the any cost function as deemed appropriate for device and application at hand.

Using the approximation method for obtaining a feasible queuing policy in Section III, we choose the tangent at $x_0 = 0.2$; we set $\delta = 0.001$ in the EFMMC framework.

Fig. 3 shows the shape of the target stationary distribution π^* and π as the results of steps 1 and 2 in Section IV-A, respectively. As a reminder π^* is the given target distribution, while π is its approximation. As seen, π^* and π are very close, indicating a very good approximation of our approach. Therefore, we should expect that the QoS produced by the approximation should not be different too much from the given QoS.

In addition, Fig. 5 shows that total variance distances of the queuing policies produced by the EFMMC and FMMC framework over the number of time steps. The total variance distance denoted the similarity between the stationary distributions produced by the two queuing policies to the given stationary distribution. As seen, EFMMC framework has a faster convergence rate as expected. However, it is clear from the Fig. 5 that over time, the FMMC framework produces the exact given distribution $\pi*$ while the EFMMC fails to do so. *Despite of this, it is important to note that a faster convergence rate is sometimes preferred. Especially, it is useful in non-stationary settings when network environment changes*

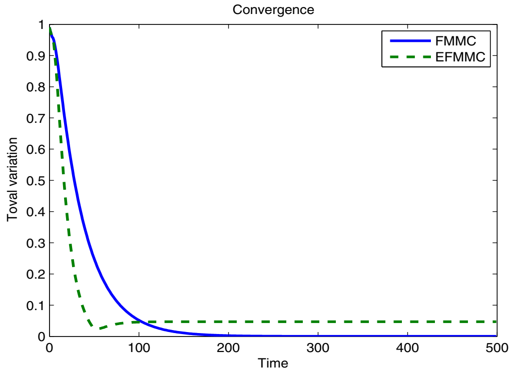

Figure 5: Comparison of the convergence times in two cases

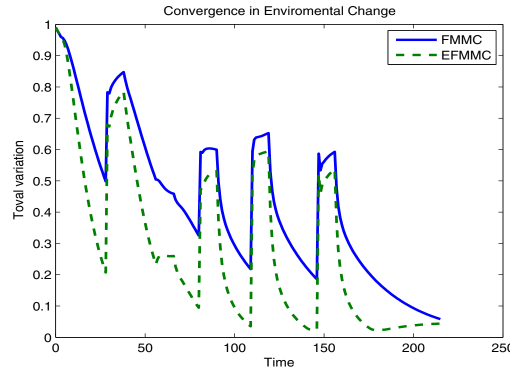

Figure 6: Convergence of the system during environmental change

rapidly such that there is no time for an optimal queuing policy to converge to the optimal one. In other words, by the time the queuing policy produces a stationary distribution, that stationary distribution is no longer the optimal one since the network conditions have changed. In such situations, it is better to implement a policy that converges fast, although to an approximate target distribution.

To illustrate this point, Fig. 6 shows the total variance between the current distributions produced by the FMMC and EFMMC frameworks, and the target stationary distribution in a non-stationary environment. The non-stationary environment is simulated based on the bursty traffic Poisson patterns with $\lambda = 30$. Specifically, in addition to the regular traffic, there are 5 bursts of packets arriving at the queue. On average, the time duration between these bursts are 30 time slots. As shown in Fig. 6, both curves have spikes when the bursts of packets arrive. As seen, the network conditions changes rapidly that prevents the queuing policies produced by both frameworks FFMC and EFMMC from approaching the their stationary distributions (i.e, the curve approaching zero for FMMC and ϵ for EFMMC). On the other hand, the queuing policy based on EFMMC framework is better than that of FMMC since it produces as close as possible to the target distribution quickly before the network conditions change.

One quantitative measure for improvement of the EFMMC framework is to measure the areas under the curve for each framework. Since the area measures the total difference between the target distribution and the distributions produced by the queuing policies over time, the smaller the area is the better. As seen EFMMC is a clearly a better policy in such rapid-changing network conditions.

V. Conclusion

In this paper, we have proposed a framework for finding fast queuing policies that can provide both flexible QoS requirements as well as optimize for a given objective. Our framework is developed based on convex optimization and mixing time theory for Markov chains. We recommend that when the network conditions change slowly, it is better to used the proposed FMMC framework for finding the queuing policy. On the other hand, when the network conditions change rapidly, it is better to employ the proposed EFMMC for finding the queuing policy. The analysis and simulation results show and verify the benefits of the proposed framework.

References

[1] L. Kleinrock, *Queueing systems, volume 1: Theory*. New York: John Wiley & Sons, 1975.

[2] J. Nagle, "On packet switches with infinite storage," *Communications, IEEE Transactions on*, vol. 35, no. 4, pp. 435 – 438, apr 1987.

[3] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *Networking, IEEE/ACM Transactions on*, vol. 6, no. 5, pp. 611 –624, oct 1998.

[4] W. Ding and B. Liu, "Rate control of mpeg video coding and recording by rate-quantization modeling," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 6, no. 1, pp. 12–20, Feb. 1996. [Online]. Available: http://dx.doi.org/10.1109/76.486416

[5] J.-C. Bolot and T. Turletti, "A rate control mechanism for packet video in the internet," in *INFOCOM*, 1994.

[6] I. S. 802.11e/D6.0, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements*, 2003.

[7] D. A.Levin, Y. Peres, and E. L.Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[8] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM Review*, vol. 46(4), pp. 667–689, December 2004.

[9] D. Nguyen-Huu, T. Duong, and T. Nguyen, "Achieving quality of service via packet distribution shaping," in *GLOBECOM*, 2012.