Network Protocol Designs: Fast Queuing Policies via Convex Relaxation

Duong Nguyen-Huu, Thai Duong, and Thinh Nguyen, *Member, IEEE*

***Abstract*—With the recent rise of mobile and multimedia applications, other considerations such as power consumption and/or Quality of Service (QoS) are becoming increasingly important factors in designing network protocols. As such, we present a new framework for designing robust network protocols under varying network conditions that attempts to integrate various given objectives while satisfying some pre-specified levels of Quality of Service. The proposed framework abstracts a network protocol as a queuing policy, and relies on convex relaxation methods and the theory of mixing time for finding the fast queuing policies that drive the distribution of packets in a queue to a given target stationary distribution. In addition, we show how to augment the basic proposed framework to obtain a queuing policy that produces ϵ-approximation to the target distribution with faster convergence time which is useful in fast-changing network conditions. Both theoretical and simulation results are presented to verify the effectiveness of the proposed framework.**

***Index Terms*—Quality of service, queuing analysis, optimization, convergence, probability distribution.**

I. Introduction

With the recent rise of mobile and multimedia applications made possible by various wireless network architectures, other considerations such as power consumption and/or Quality of Service (QoS), e.g., the requirements on minimum bandwidth, maximum jitter, delay, or loss, are becoming increasingly important factors in designing network protocols. For example, a real-time video conference application might employ a real-time video streaming protocol which is designed with emphasis on packet delay. On the other hand, a network protocol that consumes less power is preferable for smart phones. In addition, to be efficient, today network protocols must cope with the fast-changing and non-stationary characteristics of wireless channels as well as fluctuating traffic amount induced by the diversity of modern applications. Therefore, in this paper, we present a framework for customized designs of robust network protocols that achieve various objectives and requirements imposed by the heterogeneity of applications and hardware architectures under fast-changing, non-stationary environments.

The proposed framework relies on three components: (1) the abstraction of a network protocol as a queuing policy in order to allow for generalization of protocol designs as well as tractable analysis using queuing theory; (2) the optimization formulation, specifically the convex relaxation, that provides the flexibility in specifying various objectives and constraints induced by different applications and hardware architectures; (3) the theory of mixing time that helps design the protocols to promptly achieve the given objective in order to minimize the effect of the abrupt changes in the environments.

Network Protocol as Queuing Policy. Many network simulators such as NS [1] use queue to model a physical link [2]. Specifically, the packets in transit can be thought as to be in a ”queue”. The rates at which the packets being injected to the network by a sender, and being read by a receiver can be thought as the ”enqueue” and ”dequeue” rates, respectively. More abstractly, queue is also used to model higher level network protocols such as TCP. Specifically, TCP packets that have not been acknowledged can be thought as in the ”queue” or lost packets. The rates at which TCP packets being sent and read by the sending and receiving applications can be thought of as the ”enqueue” and ”dequeue” rates, respectively. The TCP ACK signals is used to control the ”enqueue” rate while the ”dequeue” often depends on the receiver’s platform. A queue can also directly model an actual physical queue residing at a wireless transmitting device such as Wi-Fi devices. In this case, the rates at which packets being injected into a physical queue by a sending application and being successfully sent out of the queue can be thought of as the ”enqueue” and ”dequeue” rates, respectively which can be controlled by the application and the MAC protocol. Thus, understanding the dynamics of packets in queues over time as a result of employing certain queuing policy/network protocol enables the system engineers to characterize and to predict various properties of the data flow such as bandwidth, packet loss, and delay.

Stationary Distribution. Central to our approach is the notion of the *stationary distribution* of packets in the queue associated with each queuing policy. The stationary distribution is important in characterizing various properties of a protocol. Stationary distribution can be used to characterize the traditional QoS metrics such as loss and delay. Also, it is an important parameter to be optimized for many objectives such as the average consumption power of the protocol. Therefore, finding a queuing policy that produces a desired stationary distribution in the fastest time is one of the main goals of the paper.

Contribution: Our contributions include a convex optimization framework for providing Quality of Service (QoS) using a *fast* queuing policy that achieves a given stationary distribution. A given stationary distribution allows for a more

Manuscript received June 19, 2013; revised October 18, 2013. The editor coordinating the review of this paper and approving it for publication was C. W. Tan.

The authors are with the School of Electrical Engineering and Computer Science, Oregon State University, Oregon, OR, 97331 USA (e-mail: {nguyendu, duong, thinhq}@eecs.oregonstate.edu).

This work is supported in part by NSF Career Award CNS-0845476.

Digital Object Identifier 10.1109/TCOMM.2013.120713.130450

0090-6778/14$31.00 © 2014 IEEE

general and precise control of various QoS requirements. In addition, we show how an even faster queuing policy can be achieved when the queuing policy only needs to produce a stationary distribution that is ϵ-close to the target stationary distribution using the theory of fast mixing Markov chain and convex optimization. The fast adaptive queuing policies are especially useful for applications in fast-changing network conditions. Finally, we show how the proposed framework can be applied to optimize for a wide range of objectives beyond the standard QoS requirements, such as power consumption.

II. Related Work

Network protocols. There exists a vast literature on network protocol designs. Typically, network protocols are designed based on a few principles, and are optimized for specific situations. Wireless network protocols such as Wi-Fi protocols are completely different from the network protocols running on a wired network such as TCP in terms of operations as well as objectives. There are also protocols tailored for multimedia transmission applications [3], [4], [5] where emphasis is on minimizing delay jitter and protocols for sensing applications with a focus on minimizing power consumption [6], [7], [8], [9], [10]. All of these protocols are typically designed to achieve some certain objectives. For example, TCP attempts to improve the bandwidth efficiency through congestion control and avoidance mechanisms. While many protocols are designed to respond quickly to changes in network conditions, they are often designed in heuristic ways. This paper provides a flexible framework for designing network protocols that achieves a wide class of objectives, and formalizes the notion of a fast response protocol via the notion of mixing time.

Network protocols for multimedia traffic. Another aspect of network protocols/queuing policies design aims at satisfying a given of QoS as specified by certain multimedia applications such as audio/video interactive and streaming applications. The underlying principle for providing QoS under a limited resource setting is to treat packets differently based on their priorities. For example, packets of flows of different priorities are classified and marked at the ingress routers in the proposed DiffServ architecture [11]. The markings are then used by the intermediate routers to determine their forwarding/queuing policies.

The same principle is also applied in local wireless area networks (WLAN). Specifically, using the MAC protocol 802.11e in the Enhanced Distributed Channel Access (EDCA) mode [12], [13], [14], [15], packets are classified into different types: Background (AC_BK), Best Effort (AC_BE), Video (AC_VI), Voice (AC_VO). The minimum and maximum contention window (CWmin, CWmax) and Arbitration Inter-Frame Space (AIFS) are the primary parameters to control the priorities for different packet types.

Another approach to provisioning flows of different priorities is to employ multiple physical or virtual queues at a router. Each queue consists of packets of the same type. A queuing policy such as simple fair queuing [16] or weighted queuing [17] is used at each transmission opportunity, to decide which of the queues whose a packet should be transmitted. Recently, there is also a number of queuing policies related to our work, but are designed for different objectives. For example, a queue can be implemented to give priority to small service requests in order to reduce the mean queue length [18]. In these types of policies, the optimal one is known as Shortest-Remaining-Processing-Time [19], [20], which shows a dramatic improvement in term of the mean response time [21], [22] [23], [24]. Unlike our work, all the above work do not study the convergence rate which plays a critical role in non-stationary environments.

III. Preliminaries

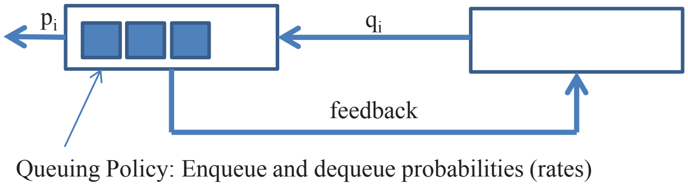
A. Queuing Policy, Stationary Distribution, and QoS

A network protocol is abstracted as a queuing policy which is governed by a tridiagonal transition probability matrix as shown in Fig. 1. The dimension N of the matrix represents the maximum length of the queue. The diagonal, left-of-diagonal, right-of-diagonal entries in the tridiagonal transition probability matrix represents the probabilities that the number of packets in the queue stays the same, decreases by one, or increases by one, respectively. We first use a simple example of a discrete-time version of the classical $M/M/1/k$ queuing model to illustrate the relationship between the stationary distribution induced by a queuing policy and QoS. Assume that at the beginning of each time step, exactly one packet arrives at the queue with probability $p = 0.4$. Otherwise, with probability $1 - p = 0.6$, no packet arrives during that entire time step. We note that p captures the current traffic conditions, i.e., how often packets arrive. Furthermore, a queuing policy is used such that at the beginning of each time step, exactly one packet is dequeued with probability $q = 0.6$. Otherwise, with probability $1 - q = 0.4$, no packet is dequeued during that entire time step. For simplicity let $k = 2$ be the maximum queue size, and a newly arrived packet is dropped if the queue is full. The dynamic of the number of packets in the queue over time can be shown to be governed by the following transition probability matrix:

$$P = \begin{pmatrix} 0.84 & 0.16 & 0 \\ 0.36 & 0.48 & 0.16 \\ 0 & 0.36 & 0.64 \end{pmatrix},$$

where P_{ij} denotes the probability that the queue will have j packets in the next time step, given that it currently has i packets with $i, j \in \{0, 1, 2\}$. As seen, P_{ij} depends on both queuing policy and traffic characteristics. Now, for each aperiodic and irreducible P, there exists a unique corresponding stationary distribution π such that $\pi^T P = \pi^T$. In this particular case, $\pi = \begin{pmatrix} 0.61 & 0.26 & 0.12 \end{pmatrix}^T$.

The stationary distribution π characterizes the long term or stationary probability of the queue occupancy. In this case, out of all the observed time slots, 61% of time the queue is empty, 27% of the time the queue has exactly one packet, and 12% of the time the queue has two packets. Knowing the distribution , the average queuing delay can be precisely calculated. One can also immediately bound the probability of dropped packets to no more than 0.12. In addition, by setting the actual duration of each time slots appropriately, one can effectively obtain the desired the bandwidth, similar to the time-scaling techniques



$$\begin{pmatrix} r_1 & q_1 & & & & & \\ p_2 & r_2 & q_2 & \cdots & & & \\ & p_3 & r_3 & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & & r_{n-2} & q_{n-2} & \\ & & & \cdots & p_{n-1} & r_{n-1} & q_{n-1} \\ & & & & & p_n & r_n \end{pmatrix}$$

Fig. 1: Queuing policy can be viewed as a tridiagonal transition probability matrix

currently used in Wi-Fi network (802.11b vs. 802.11g). In fact, any statistical measure, e.g., moments of any order can be theoretically calculated for the given stationary distribution and transition probability matrix.

As discussed, the transition probability matrix P is induced by both a queuing policy and the traffic conditions which in turn produces a stationary distribution π. Suppose the QoS requirements are given in terms of maximum average packet latency and minimum packet drop rate, then one can find a stationary distribution π that satisfies such requirements. However, there are many transition probability matrices P's that produces the same stationary distribution π. For simplicity, let us assume that the traffic is stationary, then to find the *right* desired transition probability matrix or queuing policy among all the possible queuing polices, further requirements and criteria on the desired queuing policy are needed. We discussed these issues below.

Constraints on Queuing Policy. Intuitively, for a high priority flow $\pi = (1, 0, 0)^T$ seems to be the best stationary distribution since the queue is always empty. However, this implies that a packet is always dequeued at every time slot. This policy might not be possible or optimal due to several reasons. For example, let us consider a wireless network consisting of multiple nodes. First, if an application does not require much throughput, then sending packets all the time consumes more power than necessary. Second, if every node in the wireless network implements the same greedy queuing policy, then collisions will happen constantly, resulting in low overall throughput. Thus, the transition probability matrix must be selected from a pre-specified class of transition probability matrices that gives rise to reasonable queuing policies for the given settings. These types of constraints is an input to our optimization framework.

Fastest Queuing Policy. The theory of Markov chain shows that if we apply the same queuing policy over many time steps, the distribution of packets in the queue will converge to a unique stationary distribution corresponding to a stochastic, aperiodic and irreducible matrix P. Formally, let ν be any initial distribution of packets in the queue, then

$$\lim_{n \to \infty} \nu^T P^n = \pi^T, \tag{1}$$

where n is the number of time steps. As seen, π can be obtained approximately using the same queuing policy after some sufficiently large number of time steps. Among all the P that have the same π, we want the queuing policy that drives the distribution of packets in the queue to the desired stationary distribution at the fastest rate. This is especially useful when the network conditions change and thus fast adapting queuing policy is preferable.

That said, our approach to network protocol design is based on (1) stationary distribution and (2) constraint on queuing policies rather the classic approach to protocol design based on packet loss and congestion control. In addition, we do not know whether the network conditions will change significantly in the next second or not, so our approach is to find the protocol that achieves a pre-specified objective as quickly as possible, given the current conditions.

B. Mixing Time and Spectral Gap

In order to quantify "fast" queuing policy, we first 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

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

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. A non-zero vector v_i is called a right (left) eigenvector of a square matrix P if there is a scalar λ_i such that: $Pv_i = \lambda_i v_i$ or $(v_i^T P = \lambda v_i^T)$. The scalar λ_i is said to be an eigenvalue of P. If P is a stochastic matrix, then $|\lambda_i| \leq 1, \forall i$. Denote the set of eigenvalues in non-increasing order:

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

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

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

We also 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

$$P_{ij}\pi(i) = P_{ji}\pi(j). \tag{3}$$

Proposition 1: For any discrete-time Markov chain with tridiagonal stochastic transition matrix, the chain is reversible.

Proof: See [25]. ■

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): [25] 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

$$t_{mix}(\epsilon) \leq \frac{1}{1-\mu(P)} \log\big(\frac{1}{\epsilon\pi_{min}}\big). \tag{4}$$

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 rate which will be the topic in the next section.

IV. Robust Queuing Policies/Network Protocols via Convex Relaxation Optimization

In this section, we present a number of convex relaxation optimization formulations for finding tridiagonal transition probability matrix with fast mixing rate that achieves a given target stationary distribution. Based on this, we present an augmented framework for finding fast queuing policies that are optimized any convex objective in stationary distribution.

A. Fast Mixing Tridiagonal Matrix for a Given Stationary Distribution

We assume that a stationary distribution is given. The goal is to find a tridiagonal transition probability matrix with the fastest mixing rate. It was shown in [26] that

$$\mu(P) = ||D_\pi^{1/2} P D_\pi^{-1/2} - \sqrt{\pi}(\sqrt{\pi})^T||_2, \tag{5}$$

where π denotes the stationary distribution of P, D_π denotes the square diagonal matrix whose diagonal entries are taken from each elements of π, and $||.||$ denote l_2-induced matrix norm. In (5), P must be reversible. Furthermore, $\mu(P)$ is a convex function in P with the following definition.

Definition 5 (Convex function): A function $f : \mathbf{R^n} \to \mathbf{R}$ is said to be convex if

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y)$$

for all $x, y \in \mathbf{R^n}$ and all $\alpha, \beta \in \mathbf{R}$ with $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.

Our first convex optimization is: given a target stationary distribution, design the fastest chain with transition matrix (P) that drives the chain from any initial distribution to the target stationary distribution. We consider the fast chain is the one with the maximum "mixing rate" $\mu(P)$, rather than the "mixing time" $t_{mix}(\epsilon)$. This is due to the fact that we cannot order functions $t_{mix}(\epsilon)$ by their values. On the other hand, it is straightforward to order the functions via their highest exponents or mixing rates $\mu(P)$. This problem of finding the chain with fastest mixing rate was first formulated broadly in [26] as:

Problem 1 - FMMC (Fastest Mixing Markov Chain).

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

The objective function is SLEM. The first constraint ensures P is a stochastic matrix. The second constraint is for reversibility. The third constraint is imposed by limitations of certain settings of the chain. The solution of the problem (if exists) is a transition matrix P_{opt} which has the smallest SLEM, resulting fastest convergence time to the given target distribution π^*. However, these constraints, especially the third constraint, can be restricted that given a stationary distribution π^*, there might not be a P that simultaneously satisfies all the constraints and produces the desired stationary distribution. For example, consider a queuing policy, if one restricts the queuing policy to always send packets at some constant rate (q) regardless of how many packets in the queue, then there is less flexibility in producing the desired π^*. In addition, in many settings, finding a queuing policy that produces a stationary distribution that is within some small ϵ of the target stationary distribution, but has faster convergence time might be preferable. This is especially useful when network conditions change quickly. Based on this, we propose the following optimization problem (P2):

Problem 2.

$$\text{Minimize}_{P,\pi} \ ||D_\pi^{1/2} P D_\pi^{-1/2} - \sqrt{\pi}(\sqrt{\pi})^T||_2$$
$$\text{Subject to}: \begin{cases} P\mathbf{1} = \mathbf{1} \\ ||\pi^* - \pi||_2 \leq \epsilon \\ D_\pi P = P^T D_\pi \\ \text{Other convex constraints on } P. \end{cases} \tag{7}$$

The optimization variables in (P2) are both P and π. Unfortunately, (P2) is non-convex in P and π. Therefore, we propose the following convex problem (P3) to find an approximate solution for (P2).

Problem 3 - EFMMC (Extended Fastest Mixing Markov Chain).

$$\text{Minimize}_P \ ||D_{\pi^*}^{1/2} P D_{\pi^*}^{-1/2} - \sqrt{\pi^*}(\sqrt{\pi^*})^T||_2$$
$$\text{Subject to}: \begin{cases} P\mathbf{1} = \mathbf{1} \\ ||{\pi^*}^T P - {\pi^*}^T||_2 \leq \delta \\ P \text{ is reversible.} \\ \text{Other convex constraints on } P. \end{cases} \tag{8}$$

Unlike (P2), P is the only optimization variable in (P3). As a result, Problem (P3) is now convex. Note that the constraint $DP = P^T D$ in Problem 1 is convex for a given π. However, when π is another optimization variable as in the Problem 3, the constraint $DP = P^T D$ is not convex in P and π. Therefore, we cannot directly use $DP = P^T D$ for Problem 3. Instead, one can specify the reversibility by other means without involving π explicitly. For example, Proposition 1 shows that any tridiagonal matrix is reversible. Thus, we can enforce the tridiagonal structure directly as $P_{ij} = 0$ for $|i - j| > 2$. These constraints are convex in P. In general, there might be other forms of P beyond tridiagonal that are reversible.

We note that there should likely exist a matrix P for which $\pi^T P = \pi^T$ as there are $2|\Omega|$ equations (including stochastic transition requirement) with $|\Omega|^2$ variables (entries) in P. However, one can imagine that the constraints on P imposed by a system can be so restrictive, e.g., a queuing policy cannot send above certain rate and cannot reduce the incoming rate. Then there isn't a solution that satisfies the requirement, e.g., very small average packet loss rate. To make sure that there exists a solution P, one might need to solve these equations and constraints. We do not know an analytic way to do so. On the other hand, from an algorithmic viewpoint, if there does not exist a solution, most convex solvers will return infeasible result after many iterations. In such cases, one can enlarge the feasible region by increasing ϵ that controls the difference between the target π^* and a feasible π.

One issue to consider is how to pick δ in the constraint $||\pi^{*T} P - \pi^{*T}||_2 \leq \delta$, so that the solution to (P3) satisfies all the constraints in (P2). Specifically, we want to determine the bound on the value of δ to guarantee that the constraint $||\pi^* - \pi||_2 \leq \epsilon$ in (P2) is satisfied. We have the following propositions.

Proposition 2: For any irreducible aperiodic reversible P, we have:

$$||\pi^* - \pi||_2 \leq \frac{\pi_{max}^{1/2}}{\pi_{min}^{1/2}} \frac{||\pi^{*T} P - \pi^{*T}||_2}{1 - \lambda_2}. \tag{9}$$

Proof: See Appendix. ■

From Proposition 2, it is straightforward to see that if we pick $\delta \leq \epsilon \sqrt{\frac{\pi_{min}}{\pi_{max}}} (1 - \lambda_2)$, then $||\pi^* - \pi||_2 \leq \epsilon$. On the other hand, we cannot possibly know π_{min}, π_{max}, and λ_2 without knowing P first. However, one often can find some upper and lower bounds on these quantities based on the structure of the class of the transition matrices. For example, one can bound λ_2 via the conductance obtained by examining the corresponding graph $G(V, E)$ [27].

So far the proposed framework is applicable for a general class of reversible matrices. Now we show the results applicable to tridiagonal matrices, i.e., queuing policies. We have the following results.

Proposition 3: Let P be a tridiagonal matrix with $\alpha \leq P_{ij} \leq \beta; (0 < \alpha < \beta < 1)$ for all (i, j) in the off-diagonal line, we have

$$\begin{cases} \pi_{min} \geq \alpha^{|\Omega|-1} \\ \pi_{max} \leq \beta \\ \lambda_2 \leq 1 - 2\alpha^{2|\Omega|} \end{cases}$$

From the queuing policy's perspective , the values of α and β can model the system limitations on maximum sending and receiving rates under certain traffic conditions due to power consumption or other constraints.

Proof: See Appendix. ■

Using Proposition 3, the following corollary is obtained for selecting the right δ based on ϵ.

Corollary 2: For the class of tridiagonal matrices defined in Proposition 3, pick $\delta = \epsilon \frac{2\alpha^{(5|\Omega|-1)/2}}{\beta^{1/2}}$ we will guarantee that

$$||\pi^* - \pi||_2 \leq \epsilon. \tag{10}$$

We are ready to show the main result on bounding the optimal objective value of problem (P2) with that of problem (P3). Now, we have the following proposition:

Proposition 4: Let the μ_2 and μ_3 be the optimal objective values of problems (P2) and (P3), respectively. Let $\Delta = \frac{\epsilon}{\sqrt{\pi^*_{min}}}$. π^*_{min} and π^*_{max} denote the maximum and minimum entries in π^*, respectively. Then,

$$|\mu_2 - \mu_3| \leq C, \tag{11}$$

where

$$\begin{aligned} C &= \frac{\Delta(2\sqrt{\pi^*_{min}} - \Delta)}{(\sqrt{\pi^*_{min}} - \Delta)^2} + (\sqrt{\pi^*_{max}} + 2\Delta) \frac{\Delta^2}{{\pi^*_{min}}^{3/2}} \\ &+ |\Omega|\Delta(2\sqrt{\pi^*_{max}} + 3\Delta) \end{aligned} \tag{12}$$

Proof: See Appendix. ■

Proposition 4 provides both upper and lower bounds on using solution to (P3) as an approximate solution for (P2). Therefore, we can use (P3) to obtain a solution matrix P whose stationary distribution is ϵ-close to stationary π, and has faster mixing rate than that of (P1).

B. Algorithmic Solution to Proposed Framework

The proposed FMMC and EFMMC formulations are convex optimization problems for which there are various well-known algorithms to find the solutions. One such algorithm is the Gradient algorithm in which, the search direction follows the gradient direction at the current point. This method assumes that the gradient can be computed efficiently. For many problems, it is difficult to compute the gradient especially when the objective function has no derivatives at some point. Instead, another well-known algorithm called Subgradient [28] is often used. Using the Subgradient algorithm, the search direction follows the "negative" subgradient direction rather than the gradient direction. The assumption is that the subgradient, to be defined shortly, can be efficiently computed. Our algorithm is based on the subgradient method due to the difficulty of computing the gradient.

The general definition of subgradient is defined in [28]. Here we define the subgradient of $\mu(P)$ for a symmetric matrix P.

Definition 6 (Subgradient of the SLEM): A subgradient of $\mu(P)$ is a symmetric matrix G that satisfies the inequality

$$\mu(\tilde{P}) \geq \mu(P) + \mathbf{Tr}\ G(\tilde{P} - P) = \mu(P) + \sum_{i,j} G_{ij}(\tilde{P_{ij}} - P_{ij}) \tag{13}$$

for any symmetric matrix $\tilde{P}$.

Proposition 5 (Subgradient via eigenvector): Suppose P is a symmetric matrix and y is the unit eigenvector associated with $\lambda_{max}(P)$. Then the matrix $G = yy^T$ is a subgradient of $\lambda_{max}(P)$.

Proof: See Appendix. ■

Now, since P is reversible but not necessarily symmetric in our problem, we transform a reversible matrix P to the matrix $A = D_\pi^{1/2} P D_\pi^{-1/2} - \sqrt{\pi}(\sqrt{\pi})^T$. It can be shown that A is symmetric and $\lambda_{max}(A) = \mu(P)$ [25, Section 12.1]. We now can perform all the computations using the symmetric matrix A and the subgradient yy^T where y is the unit eigenvector associated with $\lambda_{max}(A)$.

Given the subgradient, we can use the standard projected subgradient algorithm given below: In Algorithm 1, α_k is the step size to control the search distance from one iteration to the next. Because the search can go out of the feasible region (i.e., violates the constraint), it is necessary to project the current location back into the feasible set. Step 5 in the algorithm ensures this by finding a point P the feasible set that is close to $\tilde{P}$. Also step 5 is another optimization problem, but is a well-known convex quadratic problem, and can be solved efficiently using convex solvers such as cvx [29]. The constraints for matrix P is different for each problem (see (6) and (8) for FMMC and EFMMC frameworks, respectively). In fact, the results in section V were obtained with the help of cvx.

Algorithm 1 Projected subgradient method

1: Start with a feasible matrix P_0 and $k := 0$
2: **repeat**
3: Find eigenvector y of P_k then compute $k := k + 1$ and $G^k = yy^T$.
4: Let $\tilde{P}_k := P_k - \alpha_k G^k / ||G^k||$ where stepsize α_k satisfies: $\alpha_k \geq 0,\ \alpha_k \to 0,\ \sum_k \alpha_k = \infty$.
5: Project $\tilde{P}$ into the feasible set to get P_{k+1} by solving the following problem:

$$\text{minimize } ||P - \tilde{P}_k|| \text{ subject to constraints on } P \quad (14)$$

6: **until** $||P_{k+1} - P_k|| \leq \epsilon$

C. Finding Fast Queuing Policies Optimized For A Given Objective Function

1) Finding feasible queuing policy: Our discussion thus far has been on finding the tridiagonal transition probability matrix P with fast mixing rate. However, a tridiagonal transition probability matrix might not produce a valid queuing policy, i.e., 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 by a queuing policy. As a result, the probabilities of a packet arriving and departing at the queue when the queue length is i, are a_i and s_i, respectively. We assume a discrete-time queuing system in which packets can only arrive and depart at the beginning of each time slot.

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.

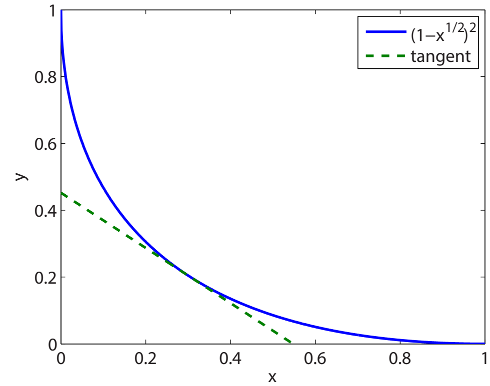


Fig. 2: Tangent at $x_0 = 0.3$ of the function $f(x) = (1 - \sqrt{x})^2$

Then 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:

$$Q = \begin{pmatrix} 1 - a_0 & a_0 & & & \\ s_1(1 - a_1) & 1 - s_1 - a_1 + 2s_1a_1 & (1 - s_1)a_1 & & \\ & \ddots & \ddots & \ddots & \\ & & & s_{|\Omega|} & 1 - s_{|\Omega|} \end{pmatrix}. \quad (15)$$

Note that for non-zero entries of each row, the left, middle, and right entries denote the probabilities that the number of packets in the queue decreases by one, stays the same, or increases by one, respectively. Now, let us compare the above matrix Q to the matrix P which is the solution obtained from the problem (P1) or (P3) above. In general, P is a tridiagonal matrix with the entries: r_i, q_i, p_i.

$$P = \begin{pmatrix} r_0 & p_0 & & & \\ q_1 & r_1 & p_1 & & \\ & \ddots & \ddots & \ddots & \\ & & & q_{|\Omega|} & r_{|\Omega|} \end{pmatrix}. \quad (16)$$

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 is possible 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 (15) and (16), we need to solve these following equations:

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

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

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

$$\iff s_i^2 - s_i(1 + q_i - p_i) + q_i = 0 \text{ for } i = 1, \ldots, |\Omega| - 1. \quad (17)$$

Let us denote s_i' and s_i'' as two roots of (17), we have:

$$\begin{cases} s_i' + s_i'' = 1 + q_i - p_i \\ s_i's_i'' = q_i. \end{cases}$$

Since $q_i, p_i \in (0,1)$ for $i = 1, \ldots, \Omega - 1$, if (s_i', s_i'') are real, at least one of s_i' or s_i'' will be in the range of $(0,1)$ which satisfies the requirements for departing probability vector s and arrival probability vector a.

Hence, in order to guarantee the existence of feasible solution of (17), we need:

$$\Delta = (1 + q_i - p_i)^2 - 4q_i \geq 0 \text{ for } i = 1, \ldots, |\Omega| - 1. \quad (18)$$

Therefore, 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 (18) by making it a convex constraint as follows.

$$(1+q_i-p_i)^2 - 4q_i \geq 0 \iff 1+q_i-p_i > 2\sqrt{q_i} \text{ since } q_i > 0$$
$$\iff (1-\sqrt{q_i})^2 > p_i. \quad (19)$$

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

Hence, (19) is equivalent to the following convex constraints:

$$a(x_0)q_i + b(x_0) > p_i \text{ for } i = 1, \ldots, |\Omega| - 1. \quad (20)$$

Now, we can incorporate the constraints in (20) to the (P1) and (P3) problems and we will guarantee their solutions to be feasible queuing policies.

2) Procedure for Optimizing a Given Objective via Queuing Policy: In this section, we describe the procedure of using our proposed framework to find a fast queuing policy that minimizes a given convex objective in the stationary distribution while satisfying other standard QoS requirements. Our approach consists of two steps. In the first step, we find a stationary distribution π^* that minimizes a given objective $f(\pi)$ subject to all the given constraints. Essentially, this step translates the QoS requirements into the target stationary distribution π. In the second step, we substitute π^* into either the FMMC or EFMMC framework with the convex constraints in (20) to find the fastest queuing policy. This fastest queueing policy is represented by P whose constraints on the entries reflect the network conditions. We give a specific example below.

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

$$\begin{cases} E[X] = \sum_{x=0}^{L} \pi(x)x < Y1 \\ E[X^2] = \sum_{x=0}^{L} \pi(x)x^2 < Y2. \end{cases}$$

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 function that might represent energy, resources that depends on queue occupancy. Now, suppose we want to minimize the total expected cost

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

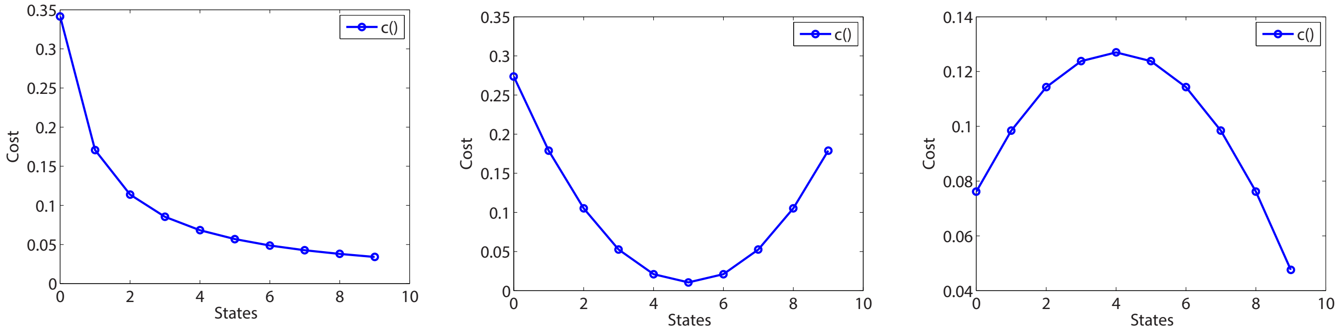
Note that $f(\pi)$ is convex in π. Then the optimization problem can be formulated as follows.

$$\text{Minimize} \sum_{x=0}^{x=L} c(x)\pi(x)$$
$$\text{Subject to}: \begin{cases} \sum_{x=0}^{L} \pi(x)x < Y1 \\ \sum_{x=0}^{L} \pi(x)x^2 < Y2 \\ \sum_{x=0}^{L} \pi(x) = 1 \\ \pi_{min} < \pi(x)\ \forall x = 0, 1, \ldots, L \end{cases} \quad (21)$$

Step 2. The solution of (21), i.e., P which 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 with fast mixing rates. Using P and the method shown in Section IV-C1, we can find the matrix Q, i.e., the enqueuing 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 while satisfying the QoS requirements.

V. Performance Evaluation of Queuing Policy Application

In this section, we present the performance evaluation of our approach using the example above with specific parameters. We assume the maximum physical queue length $L = 9$. In general, the cost function $c(x)$ is to be designed by the system engineers based the factors in consideration. To be concrete, we consider the following three basic cases. In case one, the cost function is assumed to be monotonically decreased with the queue occupancy. Intuitively, this case might appropriately model the reduction in the energy consumption with lower sending rate that leads to larger queue occupancy. Obviously, using too low a sending rate would save power but might not satisfy a given QoS requirements. In case two and case three, the cost is assumed to be a parabolic function of the queue occupancy. For case two, this cost function might appropriately model an efficient operating point of a particular hardware architecture that reduces power consumption (sending rates) while increasing efficiency by avoiding too much idle time. For case three, the cost function is approximately reversed of that in case two. Fig. 3 shows the cost functions for the three cases with their analytical expressions. The denominator in each expression is simply a normalization factor. In addition, the QoS requirements on the means $Y1$ and the second moments $Y2$ for these three cases are:
Case 1: $\{Y1 = 15; Y2 = 50; \pi_{min} = 0.01\}$; Case 2: $\{Y1 = 5; Y2 = 15; \pi_{min} = 0.01\}$; Case 3: $\{Y1 = 20; Y2 = 50; \pi_{min} = 0.01\}$. Using the approximation method for obtaining a feasible queuing policy in Section IV-C, we choose the

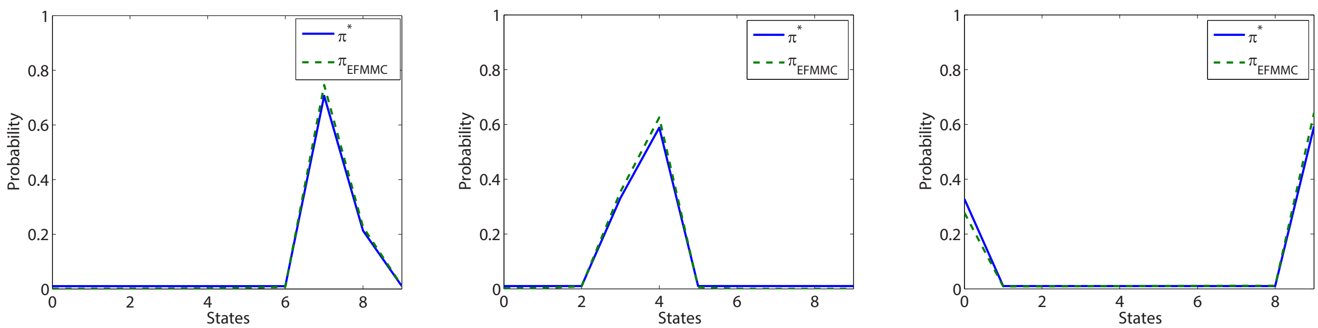


(a) Case 1: $c(x) = \frac{\frac{1}{x+1}}{\sum_{x=0}^{9} \frac{1}{x+1}}$

(b) Case 2: $c(x) = \frac{(x-5)^2+1}{\sum_{x=0}^{9} (x-5)^2+1}$

(c) Case 3: $c(x) = \frac{40-(x-4)^2}{\sum_{x=0}^{9} 40-(x-4)^2}$

Fig. 3: Cost functions $c(x)$

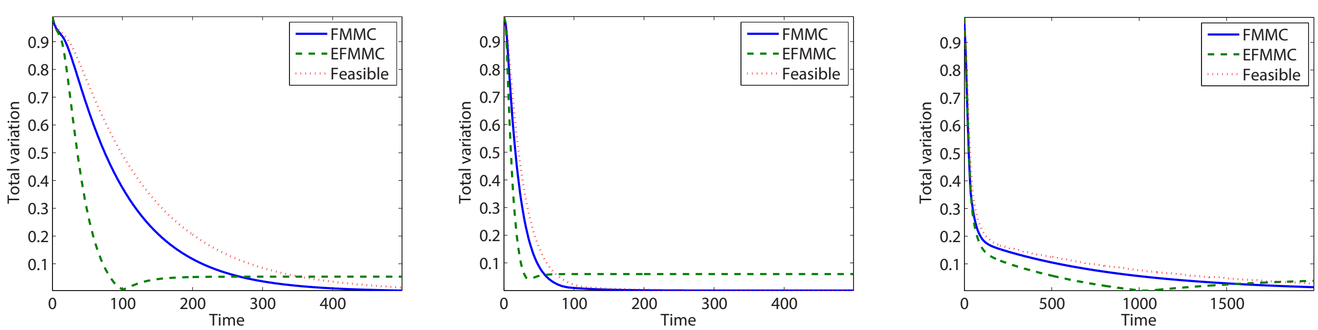


(a) Case 1

(b) Case 2

(c) Case 3

Fig. 4: Target and resulted stationary distributions



(a) Case 1

(b) Case 2

(c) Case 3

Fig. 5: Comparison of the convergence times

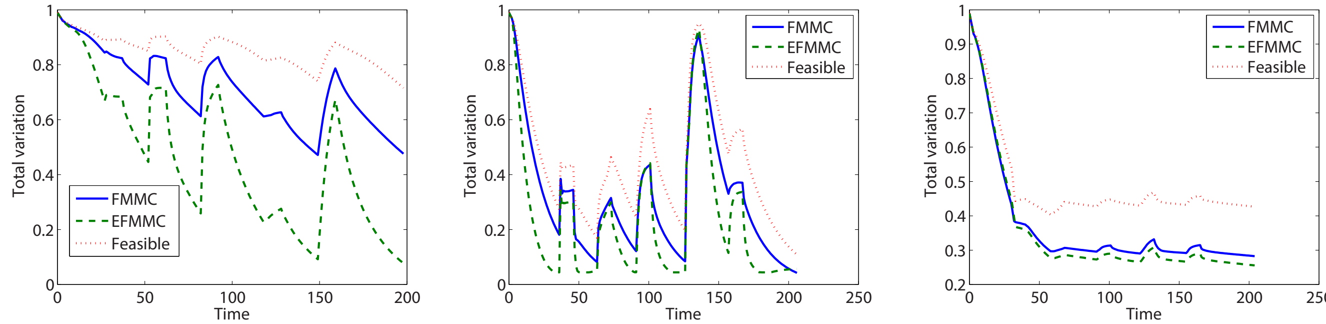
tangent at $x_0 = 0.2$; we set $\delta = 0.001$ for case 1, $\delta = 0.005$ for case 2 and $\delta = 0.0001$ for case 3 in the EFMMC framework.

To compare the effectiveness of the proposed framework, we also consider a straw-man solution which is a queuing policy in the feasible solution set of the FMMC framework and called it 'Feasible' policy. Note that it is difficult to have a meaningful comparison with other current protocols such as MAC or TCP since their objectives are quite different. Furthermore, none of the well-known protocols are optimized for fast mixing rates.

Fig. 4 shows the shape of the target stationary distribution π^* and π as the results of steps one and two in Section IV-C2, respectively. As seen, π^* and π are very close indicating a very good approximation of our approach.

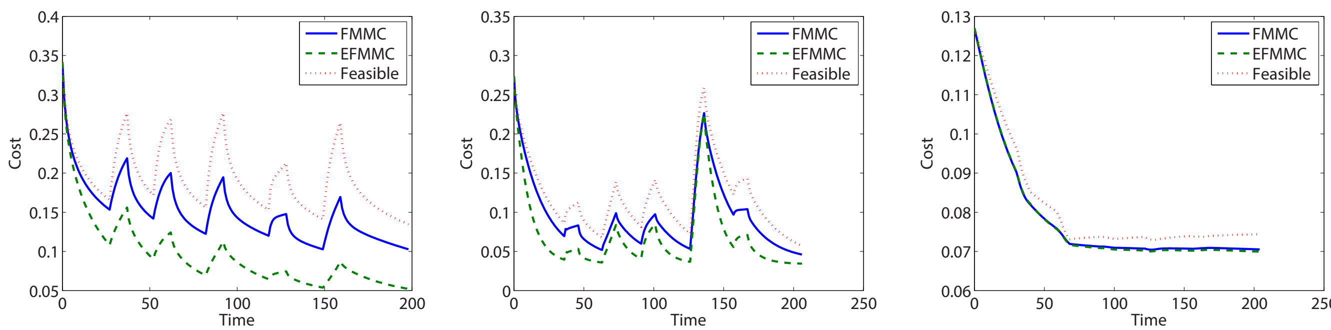
Note that we only show the performance results for three specific scenarios with three different shapes of the objective functions as seen in Fig. 4. These scenarios result in three very different π's whose masses are concentrated in on the right (Case 1), in the middle (Case 2), and on both sides (Case 3). Yet, the approximation for three cases are very good. In general, our empirical study indicate that the proposed approximation is often accurate for many typical scenarios.

Next, Fig. 5 shows the convergence rates, i.e., the total variance distance as function of number of time steps, for



(a) Case 1 (b) Case 2 (c) Case 3

Fig. 6: Total variation distances as functions of time under changing environments



(a) Case 1 (b) Case 2 (c) Case 3

Fig. 7: Costs as function of time for different queuing policies under changing environments

FMMC, EFMMC, and Feasible policies in stationary environments for all three cases. As seen, the EFMMC policy has the fastest convergence rate as expected while the Feasible queuing policy has the slowest convergence rate of all. We also note that FMMC and Feasible policies are able to drive any initial distribution to the target distribution for a given sufficient large number of time steps. This is indicated by the fact the total variance distances of the FMMC and Feasible policies approach zero with increasing number of time steps. In contrast, the EFMMC policies fails to do so. Still, the EFMMC design is preferable in fast-changing environments as will be discussed shortly.

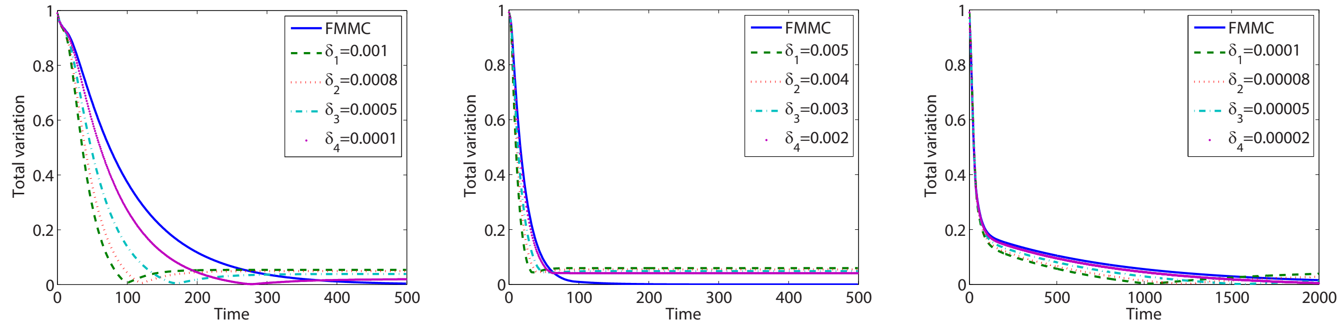
To illustrate this point, Fig. 6 shows the total variance distance between the current distributions produced by the FMMC, EFMMC and the Feasible queuing policies, and the target stationary distribution in a non-stationary environment. The non-stationary environment is simulated based on the bursty traffic 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. One can imagine that this simulation scenario models a stationary background traffic with non-stationary burst of packets as often occurred in the Internet traffic. As shown in Fig. 6, all three curves have spikes when a burst of packets arrives. This prevents the current distributions in three cases from approaching the target stationary distribution (i.e, the curves approaching zero). On the other hand, the queuing policy based on the EFMMC framework is better than that of FMMC since it produces as close as possible to the target distribution quickly.

Next, Fig. 7 shows the current expected cost of the systems by applying the Feasible queuing policy and also that of FMMC and EFMMC policies under the same non-stationary environment in the two cases. It can be seen that the expected cost induced by the EFMMC policy (the area under the curve) is the lowest of all and also it approaches the optimum cost in the fastest time. Hence, the EFMMC policy is the most efficient in all three cases.

Finally, since Proposition 4 only provides a way to pick the δ to guarantee an upper bound on the approximation, we evaluate the approximation empirically. Fig. 8 indicates that faster convergence rates can be obtained by increasing δ, and therefore ϵ. This makes sense since a larger ϵ implies relaxing the difference between π and π^*, and thus enlarging the set of P which results in finding a P with faster convergence rate.

VI. Conclusion

In this paper, we introduce a new flexible framework for designing robust network protocols under varying network conditions that attempts to integrate various given objectives while satisfying some pre-specified levels of Quality of Service. The proposed framework abstracts a network protocol as a queuing policy, and relies on the convex relaxation methods and the theory of mixing time for finding the fast queuing policies that drive the distribution of packets in a queue to a given target stationary distribution. In addition, we show how to augment the basic proposed framework in order to obtain a queuing policy that produces ϵ-approximation to the



(a) Case 1 (b) Case 2 (c) Case 3

Fig. 8: Convergence of EFMMC framework in stationary environment for various values of δ

target distribution with even faster convergence time. This fast adaptation is especially useful for networking applications in fast-changing network conditions. Both theoretical and simulation results are presented to verify the effectiveness of the proposed framework.

Appendix

Proposition 2

Proof: We assume P has n eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ and n left eigenvectors $\{v_1, v_2, \dots, v_n\}$ such that: $1 = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq -1$.

Let $\langle f, g\rangle_{\frac{1}{\pi}} := \sum_{i\in\Omega} \frac{f(i)g(i)}{\pi(i)}$ denote the inner product with respect to $\pi(i)$. Due to the reversibility of P, it can be shown that the set of eigenvectors $\{v_i\}$ forms an orthonormal basis with $\langle ., .\rangle_{\frac{1}{\pi}}$. The eigenvector corresponding to the largest eigenvalue $\lambda_1 = 1$ is equal to the stationary distribution: $v_1 = \pi$. We have: $\pi^{*T} - \pi^T = \sum_{i=1}^{n} \langle \pi^* - \pi, v_i\rangle_{\frac{1}{\pi}} v_i^T$

Since $v_i^T P = \lambda_i v_i^T$, then $(\pi^{*T} - \pi^T)(P - I) = \sum_{i=1}^{n} (\lambda_i - 1)\langle \pi^* - \pi, v_i\rangle_{\frac{1}{\pi}} v_i^T$.

Also,

$$\langle \pi^* - \pi, v_1\rangle_{\frac{1}{\pi}} = \langle \pi^* - \pi, \pi\rangle_{\frac{1}{\pi}} = \sum_{i=1}^{n} (\pi^*(i) - \pi(i)) = 0.$$

Hence,

$$||\pi^{*T} - \pi^T||_{\frac{1}{\pi}} = ||\pi^* - \pi||_{\frac{1}{\pi}} = \sqrt{\sum_{i=2} \langle \pi^* - \pi, v_i\rangle^2_{\frac{1}{\pi}}}$$

and

$$||(\pi^{*T} - \pi^T)(P - I)||_{\frac{1}{\pi}} = \sqrt{\sum_{i=2} (\lambda_i - 1)^2 \langle \pi^* - \pi, v_i\rangle^2_{\frac{1}{\pi}}}$$

Therefore:

$$||(\pi^{*T} - \pi^T)(P - I)||_{\frac{1}{\pi}} \geq \min_{i=2,\dots,n} |1 - \lambda_i| ||(\pi^* - \pi)||_{\frac{1}{\pi}}$$

$$\rightarrow ||(\pi^{*T} - \pi^T)(P - I)||_{\frac{1}{\pi}} \geq (1 - \lambda_2)||(\pi^* - \pi)||_{\frac{1}{\pi}}$$

$$\rightarrow ||(\pi^{*T} P - \pi^{*T})||_{\frac{1}{\pi}} \geq (1 - \lambda_2)||(\pi^* - \pi)||_{\frac{1}{\pi}}$$

Since for any vector x:

$$\frac{||x||_2}{\sqrt{\pi_{min}}} \geq ||x||_{\frac{1}{\pi}} \geq \frac{||x||_2}{\sqrt{\pi_{max}}}$$

Then we conclude:

$$||\pi^* - \pi||_2 \leq \frac{\pi_{max}^{1/2}}{\pi_{min}^{1/2}} \frac{||\pi^* P - \pi^*||_2}{1 - \lambda_2}$$

■

Proposition 3

Proof: Since $\pi^T P = \pi^T$, for any $1 \leq k \leq |\Omega|$ we have: $\pi_k = \sum_i \pi_i P_{i,k}$.

Also $P_{i,k} = 0$ for $|k - i| > 1$,

$$\pi_k = \pi_{k-1} P_{k-1,k} + \pi_k P_{k,k} + \pi_{k+1} P_{k,k+1}$$

where $P_{1,0} = 0$ and $P_{|\Omega|,|\Omega|+1} = 0$.

Hence for any k,

$$\pi_k < \pi_{k-1} \max P_{k-1,k} + \pi_k \max P_{k,k} + \pi_k + 1 \max P_{k,k+1}$$

$$\begin{aligned} &\rightarrow \pi_k < (\pi_{k-1} + \pi_k + \pi_{k+1})\beta < \beta \\ &\text{since } \pi_{k-1} + \pi_k + \pi_{k+1} < 1 \\ &\rightarrow \pi_{max} < \beta \end{aligned} \tag{22}$$

We see that P has the form:

$$P = \begin{pmatrix} P_{1,1} & P_{1,2} & & & & \\ P_{2,1} & P_{2,2} & P_{2,3} & & & \\ & \ddots & \ddots & \ddots & & \\ & & P_{|\Omega|-1,|\Omega|-2} & P_{|\Omega|-1,|\Omega|-1} & P_{|\Omega|-1,|\Omega|} \\ & & & P_{|\Omega|,|\Omega|-1} & P_{|\Omega|,|\Omega|} \end{pmatrix} \tag{23}$$

and P^2 has the form:

$$P^2 = \begin{pmatrix} P^2_{1,1} & P^2_{1,2} & P^2_{1,3} & & & \\ P^2_{2,1} & P^2_{2,2} & P^2_{2,3} & P^2_{2,4} & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & P^2_{|\Omega|,|\Omega|-2} & P^2_{|\Omega|,|\Omega|-1} & P^2_{|\Omega|,|\Omega|} \end{pmatrix}$$

where $P^2_{i,j}$ are entries of P^2. We see that the non-zero entries of P^2 has enlarged to one in each row compare to P and these entries has minimum value equal α^2. By induction, $P^{|\Omega|-1}$ would have all non-zero entries and the minimum entry value of $\alpha^{|\Omega|-1}$.

Since $\pi^T P^{|\Omega|-1} = \pi^T$, for any $1 \leq k \leq |\Omega|$ we have: $\pi_k =$

$\sum_i \pi_i P_{i,k}^{|\Omega|-1}$ where $P_{i,j}^{|\Omega|-1}$ denote the entry of row i and column j of matrix $P^{|\Omega|-1}$. Hence,

$$\pi_k \geq \sum_i \pi_i \min P_{i,k}^{|\Omega|-1} = \sum_i \pi_i \alpha^{|\Omega|-1} = \alpha^{|\Omega|-1}$$

$$\rightarrow \pi_{min} \geq \alpha^{|\Omega|-1} \quad (24)$$

Let $Q(S, S^C) = \sum_{i \in S; j \in S^C} \pi(i) P_{ij}$ for any subset S in state space Ω and S_c is complement set of S. By definition of Conductance [25], we have: $\Phi_* = min\{\Phi_S : S \in \Omega; \pi(S) \leq 1/2\}$, where $\Phi_S = \frac{Q(S,S_c)}{\pi(S)}$ for any subset S of state space and $\pi(S) = \sum_{i \in S} \pi(i)$.

Now, we can have a lower bound on $Q(S, S^C)$ for any subset S:

$$\begin{aligned} Q(S,S^C) &= \sum_{i \in S; j \in S^C} \pi(i) P_{ij} \\ &\geq \pi_{min} \min P_{ij} \geq \alpha^{|\Omega|-1} \alpha = \alpha^{|\Omega|}. \end{aligned} \quad (25)$$

Hence, Conductance $\Phi_* \geq \frac{\pi_{min}\alpha}{1/2} = 2\alpha^{|\Omega|}$.

Also, for a reversible Markov chain, let $\gamma = 1 - \lambda_2$ then $\frac{\Phi_*^2}{2} \leq \gamma \leq 2\Phi_*$ where Φ_* is Conductance of the chain. Therefore,

$$\frac{\Phi_*^2}{2} \leq \gamma \rightarrow \gamma \geq 2\alpha^{2|\Omega|} \rightarrow \lambda_2 = 1 - \gamma \leq 1 - 2\alpha^{2|\Omega|} \quad (26)$$

■

Proposition 4

Proof: Denote a vector $s = \sqrt{\pi^*} - \sqrt{\pi}$ then $|s_i| \leq \Delta \; \forall i \in \Omega$ where $\Delta = \frac{\epsilon_\pi}{\sqrt{\pi^*_{min}}} > 0$

Using Taylor series for function $f(x) = \frac{1}{c+x}$ at point $x = 0$ in the interval $x \in (-\Delta, \Delta)$, we have:

$$\frac{1}{\sqrt{\pi_i^*}} = \frac{1}{\sqrt{\pi_i} + s_i} = \frac{1}{\sqrt{\pi_i}} - \frac{1}{\pi_i} s_i + R_i$$

where R_i is the Taylor Remainder then $|R_i| \leq \frac{1}{{\pi^*_{min}}^{3/2}} \Delta^2$. Denote R is a vector whose entries are R_i then

$$\begin{cases} D_{\pi^*}^{1/2} = D_\pi^{1/2} + D_s \\ D_{\pi^*}^{-1/2} = D_\pi^{-1/2} - D_{s/\pi} + D_R \end{cases}$$

We also denote:

$$\begin{cases} A = D_{\pi^*}^{1/2} P D_{\pi^*}^{-1/2} - \sqrt{\pi^*}(\sqrt{\pi^*})^T \rightarrow \mu_3 = ||A||_2 \\ B = D_\pi^{1/2} P D_\pi^{-1/2} - \sqrt{\pi}(\sqrt{\pi})^T \rightarrow \mu_2 = ||B||_2 \end{cases} \quad (27)$$

Then we have:

$$\begin{aligned} A &= (D_\pi^{1/2} + D_s) P (D_\pi^{-1/2} - D_{s/\pi} + D_R) \\ &- (\sqrt{\pi} + s)(\sqrt{\pi} + s)^T \\ &= B + D_s P D_\pi^{-1/2} - D_\pi^{1/2} P D_{s/\pi} \\ &- D_s P D_{s/\pi} + D_\pi^{1/2} P D_R + D_s P D_R \\ &\quad - s(\sqrt{\pi})^T - \sqrt{\pi} s^T - s s^T \end{aligned} \quad (28)$$

Since $||P|| = 1$, using sub-multiplicative property of matrix norm each element in the right side of (28) (except B) can be bound as following:

$$\begin{cases} ||D_s P D_\pi^{-1/2}|| \leq max_i |\frac{s_i}{\sqrt{\pi_i}}| = \frac{\Delta}{\sqrt{\pi^*_{min}} - \Delta} \\ ||D_\pi^{1/2} P D_{s/\pi}|| \leq max_i |\frac{s_i}{\sqrt{\pi_i}}| = \frac{\Delta}{\sqrt{\pi^*_{min}} - \Delta} \\ ||D_s P D_{s/\pi}|| \leq max_i |\frac{s_i^2}{\pi_i}| = \frac{\Delta^2}{(\sqrt{\pi^*_{min}} - \Delta)^2} \\ ||D_\pi^{1/2} P D_R|| \leq max_i |\sqrt{\pi_i}||R_i| \\ \rightarrow ||D_\pi^{1/2} P D_R|| \leq (\sqrt{\pi^*_{max}} + \Delta) \frac{\Delta^2}{{\pi^*_{min}}^{3/2}} \\ ||D_s P D_R|| \leq \max_i |s_i||R_i| = \frac{\Delta^3}{{\pi^*_{min}}^{3/2}} \\ ||s(\sqrt{\pi})^T|| \leq |\Omega| \max_i |s_i(\sqrt{\pi_i^*} - s_i)| \\ \rightarrow ||s(\sqrt{\pi})^T|| \leq |\Omega| \delta (\sqrt{\pi^*_{max}} + \Delta) \\ ||(\sqrt{\pi}) s^T|| \leq |\Omega| \max_i |s_i(\sqrt{\pi_i^*} - s_i)| \\ \rightarrow ||(\sqrt{\pi}) s^T|| \leq |\Omega| \delta (\sqrt{\pi^*_{max}} + \Delta) \\ ||s s^T|| \leq |\Omega| \max_i |s_i^2| = |\Omega| \Delta^2 \end{cases}$$

Sum up all these elements, we now have:

$$\begin{aligned} ||A - B|| \leq C &= \frac{\Delta(2\sqrt{\pi^*_{min}} - \Delta)}{(\sqrt{\pi^*_{min}} - \Delta)^2} \\ &+ (\sqrt{\pi^*_{max}} + 2\Delta) \frac{\Delta^2}{{\pi^*_{min}}^{3/2}} \\ &+ |\Omega| \Delta (2\sqrt{\pi^*_{max}} + 3\Delta) \end{aligned} \quad (29)$$

Also, $|\min ||A|| - \min ||B||| \leq \max ||A - B||$. From (27), we have:

$$|\mu_3 - \mu_2| \leq C \quad (30)$$

■

Proposition 5 The proof can be found in [26, Section 5.1]

Proof: Since P is symmetric and y is a unit eigenvector associated with $\lambda_{max}(P)$, we have $\lambda_{max}(P) = y^T P y$ and $\lambda_{max}(\tilde{P}) \geq y^T \tilde{P} y$. From these two equations, we have the desired inequality:

$$\lambda_{max}(\tilde{P}) \geq \lambda_{max}(P) + y^T(\tilde{P} - P) y \quad (31)$$

$$= \lambda_{max}(P) + \sum_{i,j} y_i y_j (\tilde{P}_{ij} - P_{ij}). \quad (32)$$

Hence, $G = yy^T$ is a subgradient of P. ■

REFERENCES

[1] M. Guizani, A. Rayes, B. Khan, and A. Al-Fuqaha, *Network Modeling and Simulation: A Practical Perspective*. John Wiley and Sons, Feb 2010.

[2] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, *Fundamentals of Queueing Theory*. John Wiley and Sons, 1961.

[3] S. Floyd, J. Padhye, and J. Widmer, "TCP friendly rate control (TFRC): protocol specification," http://tools.ietf.org/html/rfc5348. Available: http://tools.ietf.org/html/rfc5348

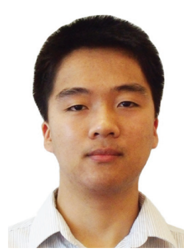
[4] I. Rhee and L. Xu, "Limitations of equation-based congestion control," in in *Proc. 2005 ACM SIGCOMM*, pp. 49–60.

[5] M. Vojnovic and J.-Y. L. Boudec, "On the long-run behavior of equation-based rate control," 2002.

[6] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. 2002 IEEE INFOCOM*, vol. 3, pp. 1567–1576.

[7] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, 2006.
[8] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," ACM Press, 2003, pp. 181–192.
[9] Y. C. Tay, K. Jamieson, and H. Balakrishnan, "Collision-minimizing CSMA and its applications to wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, pp. 1048–1057, 2004.
[10] G. Lu, B. Krishnamachari, and C. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proc. 2004 International Parallel and Distributed Processing Symposium*, pp. 224.
[11] S. Blake, D. Black, M. Carson, E. Davis, Z. Wang, and W. Weiss, "An architecture for differentiated services," in RFC2475, Dec. 1998.
[12] IEEE Std. 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.
[13] A. Grilo and M. Nunes, "Performance evaluation of IEEE 802.11e," in *Proc. 2002 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, pp. 511–517.
[14] IEEE 802.11 TGe, "HCF Ad Hoc Group Recommendation - Normative Text to EDCF Access Category," TR-02/241r0, Mar. 2002.
[15] ——, "Hybrid Coordination Function (HCF) - Proposed Updates to Normative Text of D0.1," TR-01/110r1, Mar. 2001.
[16] J. Nagle, "On packet switches with infinite storage," *IEEE Trans. Commun.*, vol. 35, no. 4, pp. 435–438, Apr. 1987.
[17] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *IEEE/ACM Transactions Networking*, vol. 6, no. 5, pp. 611–624, Oct. 1998.
[18] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal, "Size-based scheduling to improve web performance," *ACM Trans. Computer Systems*, pp. 207–233, 2003.
[19] L. Schrage, "The queue M/G/1 with feedback to lower priority queues," *Management Science*, pp. 466–474, 1967.
[20] ——, "A proof of the optimality of the shortest remaining service time discipline," *Queueing Systems*, pp. 670–690, 1968.
[21] M. Crovella and A. Bestavros, "Self-similarity in world wide web trafic: evidence and possible causes," in *Proc. 1996 ACM Sigmetrics*, pp. 160–169.
[22] M. Taqqu, W. Willinger, and A. Erramilli, *A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks*. Oxford University Press, 1996.
[23] I. Rai, E. Biersack, and G. Urvoy-Keller, "LAS scheduling approach to avoid bandwidth hogging in heterogeneous TCP networks," in *Proc. 2004 IEEE International Conference on High Speed Networks and Multimedia Communications*, pp. 179–190.
[24] ——, "Size-based scheduling to improve the performance of short TCP flows," *IEEE Network*, pp. 12–17, 2005.
[25] D. A.Levin, Y. Peres, and E. L.Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2008.
[26] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Review*, vol. 46, n o. 4, pp. 667–689, Dec. 2004.
[27] M. Jerrum and A. Sinclair, "Conductance and the rapid mixing property for markov chains: the approximation of permanent resolved," in *Proc. 1988 ACM Symposium on Theory of Computing*, ser. STOC '88, p. 235-244. Available: http://doi.acm.org/10.1145/62212.62234
[28] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," 2003, notes for EE392o, Stanford University.
[29] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," ../../cvx, Apr. 2011.

Duong Nguyen-Huu received the B.S. degree from Hanoi University of Technology, in 2009 and the M.S. from Oregon State University in 2013. He is currently a Ph.D. student with the school of Electrical Engineering and Computer Science, Oregon State University, Corvallis. His research interests are network coding, stochastic optimization, cloud computing and networking applications.



Thai Duong received the BS degree from Hanoi University of Science and Technology, in 2011 and the MS degree from Oregon State University in 2013. He is currently a Ph.D. student in Electrical and Computer Engineering at Oregon State University, Corvallis. His research interests include optimization in networking, cloud computing and wireless communications.

Thinh Nguyen received the BS degree from the University of Washington, Seattle, in 1995, and the Ph.D. degree from the University of California, Berkeley, in 2003. He is currently an associate professor with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis. He is interested in all things stochastic with applications to signal processing, distributed systems, wireless networks, network coding, and quantum walks. He has served as an associate editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the IEEE TRANSACTIONS ON MULTIMEDIA.