Q-Composer and CpR: A Probabilistic Synthesizer and Regulator of Traffic (A Probabilistic Control of Buffer Occupancy)

Sami Ayyorgun* Sarut Vanichpun§ Wu-chun Feng*

Abstract—*We present and show the correctness of two algorithms called* Q-Composer *and* CpR. Q-Composer *is a probabilistic traffic-synthesizer and* CpR *is a probabilistic traffic-regulator. Given a cumulative distribution function F,* Q-Composer *synthesizes a flow that when fed into a single-input single-output network element, the distribution of the queue-size probability at the element closely follows F.* CpR *regulates an arbitrary traffic so that when the regulated traffic (i.e. the output of* CpR*) is fed into a single-input single-output network element, the distribution of the queue-size probability at the element closely follows a prespecified cdf F.* CpR *can be viewed as a probabilistic generalization of deterministic* Leaky-bucket *regulators.* Q-Composer *and* CpR *are straightforward algorithms to implement and have applications in providing end-to-end probabilistic quality-of-service guarantees, multimedia encoding/decoding, resource allocation, and in simulation studies, beside other areas.*

1 Introduction

We present two algorithms called Q-Composer and Composer-powered-Regulator (CpR). Q-Composer is a probabilistic traffic-synthesizer and CpR is a probabilistic traffic-regulator. Input to Q-Composer, as well as to CpR, is in general a *service-curve* S (see [1, 2]) and a *cumulative distribution function* (cdf) F. For simplicity, we present Q-Composer and CpR for the special case where service-curve S in their input is of the form $S(n) = \max\{0, \rho \cdot n\}$ for an integer rate ρ; in which case, S in their input is replaced by rate ρ.

Given a ρ and an F as input, Q-Composer synthesizes a flow probabilistically that when fed into a *work-conserving server* with rate ρ (see Section 2 for the definition), the distribution of the queue-size probability at the server closely follows F. Similarly, given a ρ and an F as input and an arbitrary traffic source, CpR regulates the traffic generated by the source such that when the regulated traffic is fed into a work-conserving server with rate ρ, the distribution of the queue-size probability at the server closely follows F.

A motivation for controlling (i.e. composing or regulating) the queue-size distribution of a queue, as indicated above, arises in various networking or computing problems. The motivations included in this paper are twofold: **1)** [Providing End-to-End Probabilistic QoS Guarantees] A probabilistic characterization of network traffic had recently been introduced in [3, 4]. A special case of this characterization is given by Definition 1 in this paper. Studies in [3–5] have shown that this characterization enables a tractable analysis of end-to-end probabilistic Quality-of-Service (QoS) guarantees for both transient and steady-state regimes in communication networks. However, such a tractability requires that a characteristic of *any* traffic entering to a network is known a priori in accordance with the introduced traffic model. Q-Composer and CpR are a solution to this requirement, and make the proposed analysis framework in [3–5] viable. A traffic source can use Q-Composer intrinsically to generate flows that conform to a given characteristic in accordance with the new model, or the source can be regulated externally by CpR to achieve the same goal. **2)** [Realistic Simulation Studies] Q-Composer can be used in various simulation tools, such as ns and OPNET, to generate realistic traffic traces that induce desired queue distributions on various network elements. Such distributions might be observed a priori in a real system. Later on, a simulation study of the system might be asked for in evaluating new network protocols, where the observed queue distributions are needed to be regenerated at least for cross traffic to simulate the system realistically.

The rest of the paper is organized as follows: Section 2 provides background. Section 3 introduces Q-Composer. Section 4 introduces CpR. Section 5 provides some simulation results. Section 6 concludes the study.

2 Background

We adopt a discrete-time formulation in the context of packet networks. A *flow* is a discrete random process whose sample-paths are nondecreasing functions defined from the integers to the nonnegative integers. The value $R(n)$ of a flow R at time n denotes the cumulative number of packets that pass through a cross-section of a communication link by time n (inclusive), where the packets counted for R are specified under a certain classification. Given a flow R, the process r defined as $r(n) \triangleq R(n) - R(n-1)$ is called the *rate* of flow R. A *network element* is an input-output device or a medium that accepts packets at its input and delivers them at its output. Packets are assumed to instantaneously arrive to or depart from a network element, i.e. a whole packet can arrive instantaneously at some time k and depart at time n where $n \geqslant k$. Note that a packet can depart in the same interval in which it has arrived. The capacity $c(n)$ of a network element at time n is the maximum number of packets that it can serve/deliver at that time. A network element is said to be *work-conserving* if it serves packets at full capacity whenever it has packets to serve, unconditionally of any other criteria. Any work-conserving server is assumed to be initially empty with respect to a time origin, i.e. not storing any packet before that time, unless otherwise noted in the text. Finally, the notation z^+ stands for $\max\{0, z\}$.

*Los Alamos National Laboratory, PO Box 1663, MS B287, Los Alamos, NM 87545. E-mail: sami@lanl.gov, feng@lanl.gov

§Dept. of Electrical and Computer Eng., University of Maryland, College Park, MD 20742. E-mail: sarut@glue.umd.edu

We adopt a special case of the probabilistic traffic-characterization introduced in [4]. This special case is given by the following definition (see also [3]).

Definition 1 *A flow R is said to be bursty with a rate ρ and a bounding function f, and denoted as $R \sim (\rho, f)$, if the probability distribution of the queue-size Q of a work-conserving server with rate ρ, when fed with flow R, satisfies*

$$\mathrm{P}(Q(n) > \sigma) \leqslant f(\sigma) \quad \textit{for all } n \textit{ and for all } \sigma,$$

where f is defined from the integers to the nonnegative real numbers.

The following properties are assumed to hold for any bounding function f without loss of generality [4]:

1. f is nonincreasing, as the probability corresponding to a σ in Definition 1 is nonincreasing with σ.
2. $f(\sigma) \leqslant 1$ for all σ, as the probability of an event can not be larger than 1. Also, it is assumed for mathematical convenience that $f(\sigma) = 1$ for all $\sigma < 0$.
3. $\lim_{\sigma \to \infty} f(\sigma) = 0$, as any cdf F satisfies $\lim_{x \to \infty} F(x) = 1$.

This characterization, as well as its general form in [4], are motivated by the studies in [6–8]. The single most important difference between this characterization (in general, its general form) and the ones in [6–8] is that our characterization allows for a synthesis and a regulation method such as Q-Composer and CpR which are tight and simple. It is not clear if the characterizations in [6–8] allow for tight, as well as simple, synthesis and regulation methods. See [4] for a detailed comparison of our characterization with the ones in [6–8]. Such synthesis and regulation methods are vital for the viability of the proposed analysis methods. Q-Composer and CpR delivers that viability. See [3–5] for the tractable analysis framework enabled by this characterization for providing end-to-end probabilistic QoS guarantees.

3 Q-Composer

Q-Composer is a probabilistic traffic-synthesizer. It synthesizes a flow which conforms to a given characteristic (i.e. a rate ρ and a bounding function f) in accordance with Definition 1. Given a ρ and an f, Q-Composer generates a random integer sequence like $\langle 4, 0, 5, 2, \ldots \rangle$ which stands for generating 4 packets at time slot 1, 0 packet at time slot 2, 5 packets at time slot 3, and so on. Q-Composer utilizes another algorithm called Composer-pmf (see Appendix B). The pseudocode for Q-Composer is give below in plain text.

Q-Composer

Input: An integer rate ρ and a bounding function f.

Output: A synthetic flow R such that $R \sim (\rho, f)$.

Body: Find a probability mass function (pmf) f_A by calling Composer-pmf (see Appendix B). Input to Composer-pmf are ρ and f too, and the output is a pmf f_A defined on the integers that $f_A(a) = 0$ for all $a < 0$.

Have two random-number generators to realize the following two independent random processes:

1. $\langle X(n) : n \geqslant 1 \rangle$ is a sequence of independent and identically distributed (iid) random variables that each random variable $X(n)$ has cdf F given below
$$F(\sigma) \triangleq 1 - f(\sigma) \quad \text{for all } \sigma.$$
2. $\langle A(n) : n \geqslant 1 \rangle$ is a sequence of iid random variables that each random variable $A(n)$ has the cdf F_A corresponding to f_A.

Let r denote the rate of flow R being synthesized. Obtain flow R by the following recurrence which holds for all $n \geqslant 1$ and with boundary value $Q(0) = 0$;

$$r(n) = \min\Big\{A(n), \Big[X(n) + \rho - Q(n-1)\Big]^+\Big\} \tag{1}$$
$$Q(n) = \max\{0,\ Q(n-1) + r(n) - \rho\}. \tag{2}$$

End Q-Composer.

In the above recurrence, $Q(n)$ is equal to the queue-size at time n of an imaginary work-conserving server with rate ρ when fed with synthetic flow R; as this can be noted by (2). Also, we note that given a uniform random-number generator (e.g. *Linear Congruential Generator*) it is easy to generate random numbers whose distribution follows a given cdf, e.g. by the *inverse transform method*; see for example [9].

Q-Composer synthesizes a flow as specified in its Output, provided that Composer-pmf produces an f_A which satisfies a certain condition. The correctness of Q-Composer, as well as the condition that f_A needs to satisfy, is given by the following theorem whose proof is provided in Appendix A.

Theorem 1 *Let ρ and f be an Input to* Q-Composer*. If f_A used in* Q-Composer *satisfies the following condition*

$$(f * f_A)(\sigma+\rho) \cdot f(\sigma) \ +\ f(\sigma+\rho) \cdot F(\sigma) \ \leqslant\ f(\sigma) \quad \forall\, \sigma, \tag{3}$$

where operator '$*$' stands for convolution in Linear Systems Theory, then Q-Composer *synthesizes a flow R as specified in its Output; i.e. when R is fed into a work-conserving server with rate ρ, the queue-size Q of the server satisfies*

$$\mathrm{P}(Q(n) > \sigma) \leqslant f(\sigma) \quad \textit{for all } n \textit{ and for all } \sigma. \tag{4}$$

Composer-pmf provides a solution for f_A; input to Composer-pmf are ρ and f too, and the output is a pmf f_A satisfying condition (3). The pseudocode for Composer-pmf is given in Appendix B. A sample solution for f_A when $\rho = 10$ and $f(\sigma)$ is equal to $f_{\text{pareto}}(\sigma)$ given by (7) is shown in Figure 1. Condition (3) is trivial to verify; for this solution, both sides of (3) fall on top of each other; see Fig. 2.

4 CpR

A use of synthetic flows generated by Q-Composer is to regulate traffic with known or unknown characteristics in accordance with Def. 1, so that the regulated traffic conforms

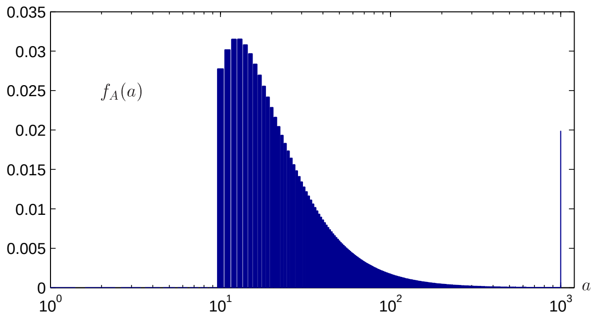

Figure 1: The pmf f_A computed by Composer-pmf for $\rho = 10$ and $f(\sigma)$ equal to $f_{\text{pareto}}(\sigma)$ given by (7).

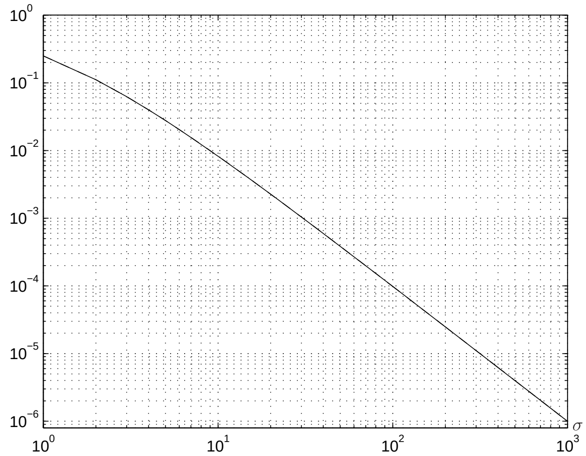

Figure 2: Both sides of the inequality in condition (3), which fall on top of each other for $\rho = 10$, $f(\sigma)$ equal to f_{pareto} given by (7), and f_A shown in Figure 1.

to a given characteristic (i.e. a rate ρ and a bounding function f). The main idea in obtaining a regulation method by using Q-Composer is simple; feed the traffic to be regulated into a buffer and then drain the buffer with a rate that is as close as possible to the rate of synthetic traffic begin generated by Q-Composer for a given ρ and f. Composer-powered-Regulator (CpR) is a such regulation method whose pseudocode is given below in plain text.

CpR

Input: An integer rate ρ, a bounding function f, a positive integer time T, an integer rate c, and a traffic source.

Output: A regulated traffic whose flow R is bursty with rate ρ and bounding function f, i.e. $R \sim (\rho, f)$.

Body: Find a probability mass function (pmf) f_A by calling Composer-pmf (see Appendix B). Input to Composer-pmf are ρ and f too, and the output is a pmf f_A defined on the integers that $f_A(a) = 0$ for all $a < 0$.

Have two random-number generators to realize the following two independent random processes:

1. $\langle X(n) : n \geqslant 1\rangle$ is a sequence of independent and identically distributed (iid) random variables that each random variable $X(n)$ has cdf F given below
$$F(\sigma) = 1 - f(\sigma) \quad \text{for all } \sigma.$$
2. $\langle A(n) : n \geqslant 1\rangle$ is a sequence of iid random variables that each random variable $A(n)$ has the cdf F_A corresponding to f_A.

Have a buffer, called B, to deposit the packets that the traffic source generates. Have a server with constant capacity c attached to buffer B, to serve the packets being queued.

Turn on the traffic source at time 1. Let r_{src} denote the rate of source traffic. Let B be initially empty; i.e. $B(0) = 0$. Read the source traffic from time 1 until time T and deposit the read packets into buffer B. Do not eject any packet from buffer B before time T. Hence, $B(T) = R_{\text{src}}(T)$.

Obtain the regulated traffic, whose rate is denoted by r, by the following recurrence: Let $Q(0) = 0$, for $n \geqslant 1$,

$$r_{\text{syn}}(n) = \min\left\{A(n), \left[X(n) + \rho - Q(n-1)\right]^+\right\},$$

$$Q(n) = \max\left\{0,\ Q(n-1) + r_{\text{syn}}(n) - \rho\right\}, \tag{5}$$

$$r(T+n-1) = \min\left\{B(T+n-1),\, r_{\text{syn}}(n),\, c\right\}, \tag{6}$$

eject (i.e. serve) $r(T+n-1)$ many packets from the head of buffer B at time $T+n-1$,

read the source traffic at time $T+n$, and deposit the read packets into buffer B,

$$B(T+n) = B(T+n-1) - r(T+n-1) + r_{\text{src}}(T+n).$$

End CpR.

The correctness of CpR is given by the following theorem whose proof is provided in Appendix C.

Theorem 2 *A flow R obtained by* CpR *is bursty with rate ρ and bounding function f given in its Input; i.e. $R \sim (\rho, f)$.*

A bounding function f entered in the Input of CpR can be determined based on the queue-size probabilities being targeted for the regulated traffic (e.g. desired overflow probabilities for a given set of queue-sizes, average queue-size, etc.). We assume for practical purposes that $f(\sigma) = 0$ for all σ larger than some sufficiently large M, e.g. $M = 10^{15}$.

A rate ρ entered in the Input of CpR can be determined in light of the following theorem which is proven in [4].

Theorem 3 *Given a flow $R \sim (\rho, f)$, where $\sum_0^\infty f(\sigma)$ is finite, the long-term average rate μ of R satisfies*

$$\mu \triangleq \limsup_{(n-k)\to\infty} \frac{\mathrm{E}[R(n) - R(k)]}{n-k} \leqslant \rho\,.$$

The long-term average rate of a traffic can be estimated by various statistical methods (e.g. The Law of Large Numbers [10]). Or, it can be known a priori by the intrinsic properties of the source generating the traffic. Let $\hat{\mu}$ denote the estimate of the long-term average rate of the traffic generated by the source that one would like to regulate. We pick an integer rate ρ larger than $\hat{\mu}$.

In CpR, we would like the capacity of buffer B (i.e. the maximum number of packets that B can store), buffer size $B(T+n-1)$ for any positive n, and rate c be "large". A "large" capacity for B would help avoid likely packet-loss due to overflow of B. A "large" c and $B(T+n-1)$ for any positive n would help get a tight regulation of traffic (i.e. the empirical bounding function of the regulated traffic would be close to the targeted bounding function f).

A holdup time T in the Input of CpR can be determined based on the tolerance of source traffic to latency; the larger the better, given the remarks in the previous paragraph.

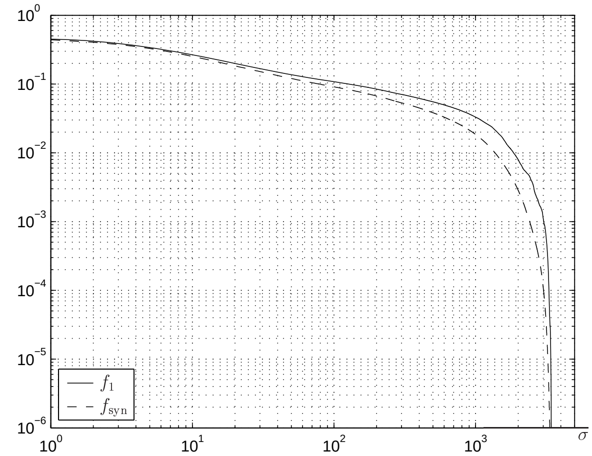

Figure 3: Plot for Sim-High.

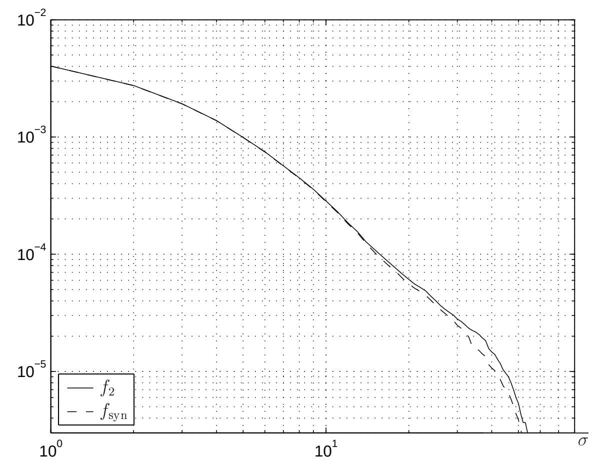

Figure 4: Plot for Sim-Moderate.

5 Simulation Results

We carried out some simulation studies to see how Q-Composer and CpR perform. In some of the simulations, we used real traffic traces collected on the institutional wide-area network of the Los Alamos National Laboratory, which we refer to as LANL-WAN. Our results show that both algorithms perform very well. The following two subsections present a part of this simulation study.

5.1 Simulations for Q-Composer

We performed four sets of simulations in which we generated synthetic traffic by using Q-Composer for various Input specifications. The bounding functions that we picked were all heavy-tailed.

Sim-High: We used a 1-hour-long real traffic trace collected on LANL-WAN to specify an Input to Q-Composer. From this trace, we obtained a flow sample-path, called R_1, by first slotting the time into 1 millisecond intervals and then counting the number of packet arrivals in each interval. The average rate of R_1 over the duration of the 1-hour collection period was 12.523 packets per time slot. For the purpose of finding a characteristic of R_1 in accordance with Def. 1, we picked a rate ρ_1 that is equal to 15. We found an estimate of the corresponding tight bounding function simply by feeding R_1 into a work-conserving server with rate 15 and then by observing the queue-size distribution of the server (see [4] for the measurability of the characterization given by Definition 1). This estimate is denoted by f_1 and shown in Figure 3. Note that the *utilization* of the server in finding a characteristic of R_1, i.e. $\frac{12.523}{15}$, was about 83%; hence the name of the simulation Sim-High.

Next, we input ρ_1 and f_1 into Q-Composer and generated 1000 synthetic traffic sample-paths each 1-hour long. We found of an estimate of the bounding function in characterizing each sample-path i as $(\rho_1, f_{\text{syn-i}})$, as indicated in the previous paragraph. We computed the point-wise average of these 1000 bounding-function estimates. This average is denoted by $f_{\rm syn}$ and is shown in Figure 3 for comparison with f_1. As we have expected by Theorem 1, $f_{\rm syn}$ is less than or equal to f_1 everywhere.

Sim-Moderate: This set of simulations was the same as Sim-High except that we used another 1-hour-long real traffic trace collected on LANL-WAN, the flow sample-path being obtained was called R_2, the average rate of R_2 was 7.7364 packets per time slot, ρ_1 was replaced with ρ_2 where $\rho_2 = 19$, and the estimate of the tight bounding function being computed in characterizing R_2 was denoted by f_2. The utilization of the work-conserving server in finding the characteristic (ρ_2, f_2) was about 41%; hence the name of the simulation Sim-Moderate. The target bounding function f_2 and the average of the estimated 1000 bounding functions of synthetic traffic being generated are shown in Fig. 4. This plot also provides a satisfactory support for Theorem 1.

Sim-Pareto: For this set of simulations, we picked a bounding function called $f_{\rm pareto}$ given below, which corresponds to a truncated Pareto distribution;

$$f_{\rm pareto}(\sigma) = \begin{cases} 1 & \text{if } \sigma < 0, \\ (\sigma+1)^{-2} & \text{if } 0 \leqslant \sigma \leqslant 10^3, \\ 0 & \text{if } \sigma > 10^3. \end{cases} \tag{7}$$

The rate ρ being picked was 10. We input ρ and $f_{\rm pareto}$ to

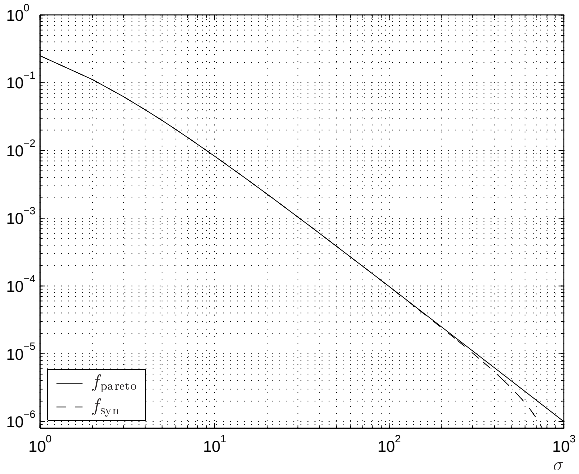

Figure 5: Plot for Sim-Pareto.

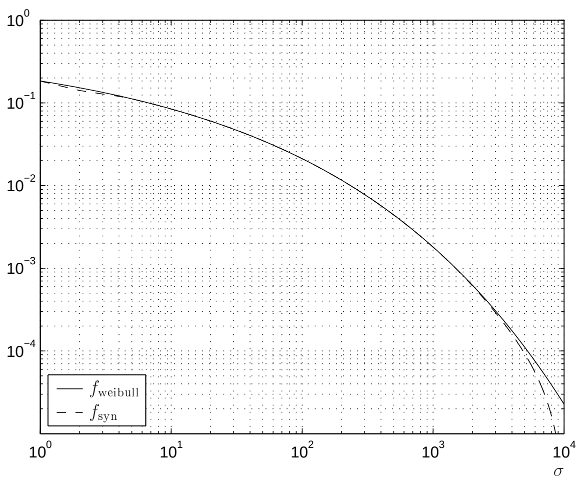

Figure 6: Plot for Sim-Weibull.

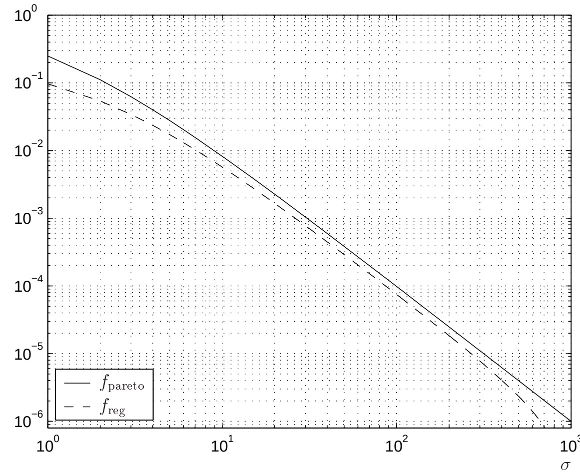

Figure 7: Plot for Sim-Reg-Pareto.

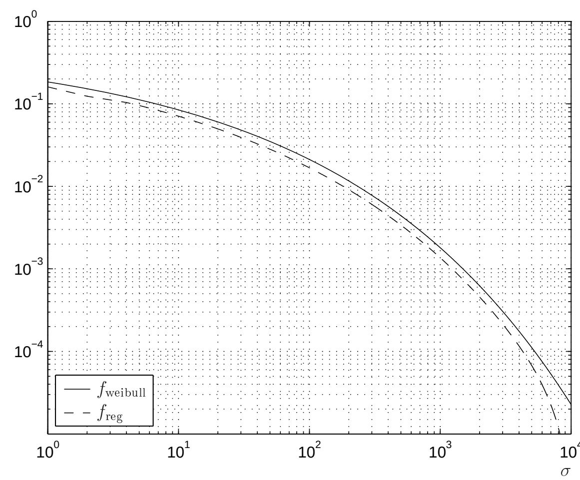

Figure 8: Plot for Sim-Reg-Weibull.

Q-Composer and generated a set of 1000 synthetic traffic sample-paths each 1-hour long. The average of the estimated bounding functions for each sample-path, i.e. f_{syn}, and the target bounding function f_{pareto} are shown in Fig. 5. This plot too provides a satisfactory support for Theorem 1.

Sim-Weibull: This set of simulations was the same as Sim-Pareto except that we picked a different bounding function called f_{weibull} given below;

$$f_{\text{weibull}}(\sigma) = \begin{cases} 1 & \text{if } \sigma < 0, \\ 0.5\, e^{-\sigma^{0.25}} & \text{if } 0 \leqslant \sigma \leqslant 10^4, \\ 0 & \text{if } \sigma > 10^4, \end{cases}$$

which corresponds to a truncated Weibull distribution. The rate ρ was 10—unchanged as noted. The average of the estimated bounding functions and the target bounding function f_{weibull} are shown in Fig. 6 which also supports Thm. 1.

5.2 Simulations for CpR

We performed two sets of simulations in which we regulated a fictitious traffic source yielding a flow whose sample-paths were all given by R_1 that was obtained in Sim-High. Recall that R_1 was derived from a real traffic trace. We regulated this traffic source by CpR so that the regulated traffic conforms to a characteristic specified as $(\,\cdot\,, \rho_{\text{reg}})$, where $\rho_{\text{reg}} = 13$, in accordance with Definition 1.

Sim-Reg-Pareto: We picked f_{pareto} given by (7) as the target bounding function for the regulated traffic to possess in its characteristic. We input $\rho = \rho_{\text{reg}}$, $f(\sigma) = f_{\text{pareto}}(\sigma)$, $T = 1$, $c = \infty$, and the fictitious traffic source described above into CpR, and regulated the source 1000 times; hence, obtained 1000 regulated flow sample-paths. The capacity of the buffer B used in CpR was chosen to be infinite. We found an estimate of the tight bounding function of each regulated flow sample-paths, as indicated in Sim-High. We computed the point-wise average of these 1000 bounding-function estimates. This average is denoted by f_{reg} and shown in Figure 7 which supports Theorem 2.

Sim-Reg-Weibull: This set of simulations was the same as Sim-Reg-Pareto except that we replaced f_{pareto} by f_{weibull} specified in Sim-Weibull. The corresponding plot is given by Figure 8 which too supports Theorem 2.

6 Conclusions and Discussions

We introduced a probabilistic traffic-synthesizer called Q-Composer and a probabilistic traffic-regulator called CpR. Both Q-Composer and CpR are straightforward to implement, and have applications in providing end-to-end probabilistic QoS guarantees, multimedia encoding/decoding, resource allocation, and in simulation studies—see Section 1. CpR can be viewed as a probabilistic generalization of deterministic *Leaky-bucket* regulators [1, 11–13].

Queue-size distributions resulted by traffic that Q-composer or CpR generate for a given (ρ, f), on other network elements (e.g. a work-conserving server with a larger rate) can easily be found by the analysis framework studied in [3, 4].

Q-Composer and CpR are not "asymptotic algorithms". That is, the inequalities that they ensure to satisfy (namely, (4) for Q-Composer and (33) for all n and σ in the proof of Theorem 2 for CpR) hold for any time n. Thus, both algorithms also enable an explicit and tractable performance analysis in transient regimes. This is an important property since most of the communication sessions in today's networks are short-lived [14]; i.e. asymptotic analysis just by themselves may not be sufficient in most cases.

Another utility of these algorithms is as follows: Determining a statistically accurate estimate of the queue-size distribution of a queue, either an asymptotic or a time-dependent one, via measurements is a labor-intensive task and is not trivial to do, especially for long-range dependent traffic. Instead of trying to measure such a distribution, we can tightly dictate the distribution that the source is desired or foreseen to induce, by using Q-Composer or CpR; hence, know a tight estimate of the distribution a priori.

An issue of concern about CpR is the queueing experienced in buffer B in externally regulating a traffic source. The combination of queueing in buffer B and in subsequent network elements is what a source traffic experiences end-to-end. However, this effect of buffer B is there to stay as long as an *unknown* traffic is decided to be regulated *externally*. The same issue also exists in regulating unknown traffic by the well-known Leaky-bucket regulator. However, this problem does *not* exist in regulating/generating traffic *internally* by using Q-Composer.

Future work includes ***a***) comparing Q-Composer with other traffic generators, ***b***) investigating other properties of interest (e.g. *self-similarity, long-range dependency*) of traffic synthesized by Q-Composer—however, queueing behavior might be considered as the single most important traffic characteristic as far as performance evaluation goes, which Q-Composer and CpR has already addressed—, and finally ***c***) studying connection admission control by using Q-Composer and CpR.

References

[1] R. L. Cruz. *Quality of Service Guarantees in Virtual Circuit Switched Networks*, IEEE Journal of Selected Areas in Communication, 13(6): 1048–1056, 1995.

[2] H. Sariowan. *A Service-curve Approach to Performance Guarantees in Integrated-service Networks.* Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of California, San Diego, 1996.

[3] Q. Yin, Y. Jiang, S. Jiang, P. Y. Kong. *Analysis on Generalized Stochastically Bounded Bursty Traffic for Communication Networks*, Proc. of the 27th IEEE Conference on Local Computer Networks (LCN), Nov. 2002.

[4] S. Ayyorgun, W. Feng. *A Systematic Approach for Providing End-to-End Probabilistic QoS Guarantees*, Proc. of the 13th IEEE International Conference on Computer Communications and Networks (ICCCN), pp. 115–122, Oct. 2004. (Also available at `http://public.lanl.gov/sami` .)

[5] S. Ayyorgun, S. Vanichpun, W. Feng. *Probabilistic QoS Guarantees over Switches and Multiplexers*, Proc. of the 42nd Annual Allerton Conference on Communication, Control, and Computing (ALLERTON), Sept. 2004. (Also available at `http://public.lanl.gov/sami` .)

[6] O. Yaron, M. Sidi. *Performance and stability of communication networks via robust exponential bounds*, IEEE/ACM Transactions on Networking, 1(3): 372–385, 1993.

[7] C. S. Chang. *Stability, Queue Length, and Delay of Deterministic and Stochastic Queueing Networks*, IEEE Transactions on Automatic Control, 39: 913–931, 1994.

[8] D. Starobinski, M. Sidi. *Stochastically Bounded Burstiness for Communication Networks*, IEEE Transactions on Information Theory, 46(1): 206–212, 2000.

[9] S. Ross. *Simulation*, Academic Press, 3rd edition, 2001.

[10] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 1*, Wiley, 3rd edition, 1968.

[11] R. L. Cruz. *A Calculus For Network Delay, Part I: Network Elements In Isolation*, IEEE Transactions on Information Theory, 37(1): 114–131, 1991.

[12] C.-S. Chang. *Performance Guarantees in Communication Networks*, Springer Verlag, April 2000.

[13] J.-Y. Le Boudec, P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, Lecture Notes in Computer Science-2050, Springer, 2002.

[14] K. Park, W. Willinger (editors). *Self-Similar Network Traffic and Performance Evaluation*, Wiley, 2000.

A Correctness of Q-Composer

The correctness of Q-Composer is given by Theorem 1. We use the following lemma in proving Theorem 1.

Lemma 1 *Let Y and Z be any two independent random variables, and let W be a nonnegative random variable independent from both Y and Z. The random variable V defined as $V \triangleq \min\{Y + W, \max\{Y, Z\}\}$ satisfies*

$$\mathrm{P}(V > v) = \mathrm{P}(Y + W > v) \cdot \mathrm{P}(Z > v) + \mathrm{P}(Y > v) \cdot \mathrm{P}(Z \leqslant v) \quad \textit{for all } v.$$

Proof: The following equalities hold for any v;

$$\mathrm{P}(V > v) = \mathrm{P}\Big(\min\{Y + W, \max\{Y, Z\}\} > v\Big)$$

$$
\begin{aligned}
&= \mathrm{P}\Big(\{Y+W>v\} \cap \{\max\{Y,Z\}>v\}\Big) \\
&= \mathrm{P}\Big(\{Y+W>v\} \ \cap \ (\{Y>v\} \cup \{Z>v\})\Big) \\
&= \mathrm{P}\Big((\{Y+W>v\} \cap \{Y>v\}) \ \cup \\
&\qquad\qquad (\{Y+W>v\} \cap \{Z>v\})\Big) \\
&= \mathrm{P}\Big(\{Y>v\} \ \cup \ (\{Y+W>v\} \cap \{Z>v\})\Big) \\
&= \mathrm{P}(Y>v) + \mathrm{P}\Big(\{Y+W>v\} \cap \{Z>v\}\Big) - \\
&\qquad \mathrm{P}\Big(\{Y>v\} \cap \{Y+W>v\} \cap \{Z>v\}\Big) \\
&= \mathrm{P}(Y>v) \ + \ \mathrm{P}(Y+W>v) \cdot \mathrm{P}(Z>v) \ - \\
&\qquad \mathrm{P}\big(\{Y>v\} \cap \{Z>v\}\big) \\
&= \mathrm{P}(Y>v) \ + \ \mathrm{P}(Y+W>v) \cdot \mathrm{P}(Z>v) \ - \\
&\qquad \mathrm{P}(Y>v) \cdot \mathrm{P}(Z>v) \\
&= \mathrm{P}(Y+W>v) \cdot \mathrm{P}(Z>v) \ + \ \mathrm{P}(Y>v) \cdot \big(1-\mathrm{P}(Z>v)\big) \\
&= \mathrm{P}(Y+W>v) \cdot \mathrm{P}(Z>v) \ + \ \mathrm{P}(Y>v) \cdot \mathrm{P}(Z \leqslant v) . \qquad \blacksquare
\end{aligned}
$$

Proof of Theorem 1:

Suppose that Composer-pmf produces an f_A which satisfies (3). Let R be a flow synthesized by Q-Composer for the given Input. Feed R into a work-conserving server with rate ρ, where the time origin is at 1. Let G denote the output flow of this server. The proof follows by mathematical induction on n.

Basis: Statement (4) clearly holds for all n less than 1; since i) $Q(n)=0$ for all $n<1$, which holds by the convention in this text that any work-conserving server is initially empty and that the time origin is at 1, and ii) for any bounding function f, $f(\sigma)=1$ for all $\sigma<0$ and $f(\sigma)\geqslant 0$ for any σ.

Induction Step: Suppose that statement (4) holds for all n less than some positive integer m—this is true by the induction *Basis* which corresponds to $m=1$. Next, we show that (4) holds also for $n=m$.

Queue-size $Q(m)$ of the work-conserving server that is fed with synthetic flow R is given by

$$
Q(m) = \max\big\{0,\, Q(m-1)+r(m)-\rho\big\}. \qquad (8)
$$

The above equality holds by how a work-conserving server is defined to work. Note that $Q(m)$ given by this equality is identical to $Q(n)$ in (2) for $n=m$.

Using (8), we get for all σ that

$$
\begin{aligned}
\mathrm{P}(Q(m)>\sigma) &= \mathrm{P}\Big(\max\big\{0,\, Q(m-1)+r(m)-\rho\big\} > \sigma\Big) \\
&= \mathrm{P}\Big(\{0>\sigma\} \cup \big\{Q(m-1)+r(m)-\rho>\sigma\big\}\Big) \\
&= \begin{cases} 1 & \text{if } \sigma<0, \\ \mathrm{P}\big(Q(m-1)+r(m)-\rho>\sigma\big) & \text{else.} \end{cases} \qquad (9)
\end{aligned}
$$

To determine the probability on the right-hand-side (rhs) in (9), we substitute $r(m)$ given by (1) and manipulate the involved random variable as

$$
\begin{aligned}
&Q(m-1)+r(m)-\rho \\
&= Q(m-1) + \min\Big\{A(m),\, \Big[X(m)+\rho-Q(m-1)\Big]^+\Big\} - \rho \\
&= \min\Big\{A(m)+Q(m-1)-\rho, \\
&\qquad\qquad \Big[X(m)+\rho-Q(m-1)\Big]^+ + Q(m-1)-\rho\Big\} \\
&= \min\Big\{Q(m-1)-\rho+A(m),\ \max\{Q(m-1)-\rho, X(m)\}\Big\}.
\end{aligned}
$$

Using Lemma 1 with the following substitutions;

$$
Y := Q(m-1)-\rho\,, \qquad W := A(m)\,, \qquad Z := X(m)\,,
$$

we get

$$
\begin{aligned}
&\mathrm{P}\big(Q(m-1)+r(m)-\rho>\sigma\big) \\
&= \mathrm{P}\big(Q(m-1)-\rho+A(m)>\sigma\big) \cdot \mathrm{P}(X(m)>\sigma) + \\
&\qquad \mathrm{P}\big(Q(m-1)-\rho>\sigma\big) \cdot \mathrm{P}(X(m)\leqslant\sigma) \\
&= \mathrm{P}\big(Q(m-1)+A(m)>\sigma+\rho\big) \cdot \mathrm{P}(X(m)>\sigma) + \\
&\qquad \mathrm{P}\big(Q(m-1)>\sigma+\rho\big) \cdot \mathrm{P}(X(m)\leqslant\sigma) \\
&= \mathrm{P}\big(Q(m-1)+A(m)>\sigma+\rho\big) \cdot f(\sigma) + \\
&\qquad \mathrm{P}\big(Q(m-1)>\sigma+\rho\big) \cdot F(\sigma). \\
&\leqslant \mathrm{P}\big(Q(m-1)+A(m)>\sigma+\rho\big) \cdot f(\sigma) + f(\sigma+\rho) \cdot F(\sigma), \qquad (10)
\end{aligned}
$$

where in obtaining the last line from the previous, we utilized the induction hypothesis.

Manipulate the probability on the rhs in (10) as;

$$
\begin{aligned}
&\mathrm{P}\big(Q(m-1)+A(m)>\sigma+\rho\big) \\
&= \sum_{a:f_A(a)>0}^{\infty} \mathrm{P}\Big(Q(m-1)+A(m)>\sigma+\rho \,\Big|\, A(m)=a\Big) \times \\
&\qquad \mathrm{P}\big(A(m)=a\big) \\
&= \sum_{a:f_A(a)>0}^{\infty} \mathrm{P}\Big(Q(m-1)>\sigma+\rho-a\Big) \cdot f_A(a)
\end{aligned}
$$

utilizing the induction hypothesis one more time, we get

$$
\begin{aligned}
&\leqslant \sum_{a:f_A(a)>0}^{\infty} f(\sigma+\rho-a) \cdot f_A(a) \\
&= \sum_{a=0}^{\infty} f(\sigma+\rho-a) \cdot f_A(a) \\
&= (f * f_A)(\sigma+\rho), \qquad (11)
\end{aligned}
$$

recall that operator '$*$' stands for convolution in Linear Systems Theory.

Using bound (11) in (10), we get

$$
\begin{aligned}
&\mathrm{P}\big(Q(m-1)+r(m)-\rho>\sigma\big) \leqslant \\
&\qquad (f * f_A)(\sigma+\rho) \cdot f(\sigma) \ + \ f(\sigma+\rho) \cdot F(\sigma).
\end{aligned}
$$

Using the above bound in (9), we have

$$P(Q(m) > \sigma) \leqslant \begin{cases} 1 & \text{if } \sigma < 0, \\ (f * f_A)(\sigma + \rho) \cdot f(\sigma) \;+\; f(\sigma + \rho) \cdot F(\sigma) & \text{else.} \end{cases}$$

Finally, since f_A is chosen to satisfy condition (3) and $f(\sigma) = 1$ for all $\sigma < 0$, the above inequality implies

$$P(Q(m) > \sigma) \leqslant f(\sigma) \quad \text{for all } \sigma. \qquad \blacksquare$$

B Composer-pmf: A Solution For f_A

We present a solution for f_A that satisfies condition (3). The algorithm that we come up with to find the solution is called Composer-pmf whose pseudocode is given at the end of this section. The correctness of Composer-pmf is shown by Steps 1, 2, and 3 preceding the pseudocode.

To begin with, we note by the following lemma that an f_A satisfying condition (3) always exists.

Lemma 2 *There exists an f_A which satisfies condition (3) for any given ρ and f.*

Proof: Let $f_A(a) = \delta(a - \rho)$ where δ is the unit sample function; i.e. $\delta(u) = 1$ if $u = 0$, and 0 otherwise. For this particular f_A, the convolution in condition (3) becomes equal to $f(\sigma)$; this is shown below,

$$\begin{aligned}(f * f_A)(\sigma + \rho) &= \sum_{a=0}^{\infty} f(\sigma + \rho - a) \cdot f_A(a) \\ &= f(\sigma + \rho - \rho) \cdot f_A(\rho) \;=\; f(\sigma)\,.\end{aligned}$$

Using the above equality, we have

$$\begin{aligned}&(f * f_A)(\sigma + \rho) \cdot f(\sigma) \;+\; f(\sigma + \rho) \cdot F(\sigma) \\ &\quad = f(\sigma) \cdot f(\sigma) \;+\; f(\sigma + \rho) \cdot F(\sigma) \\ &\quad \leqslant f(\sigma) \cdot f(\sigma) \;+\; f(\sigma) \cdot F(\sigma) \;=\; f(\sigma) \cdot \big[f(\sigma) + F(\sigma)\big] \\ &\quad = f(\sigma).\end{aligned}$$

the inequality above holds since f is nonincreasing and ρ is nonnegative. ■

The trivial solution given in the proof of Lemma 2 is not necessarily the solution that we would use in Q-Composer. A "better" solution can be found by applying a technique in the σ-domain, which is discussed in this section. Specifically, we will find a solution which also satisfies another condition, namely condition (19) pointed out later in the text, in addition to condition (3).

Let us first give a very simple fact about the convolution in condition (3) by the following lemma.

Lemma 3 *For any σ, $(f * f_A)(\sigma) \leqslant 1$.*

Proof: For any σ, we have

$$(f * f_A)(\sigma) \;=\; \sum_{a=0}^{\infty} f(\sigma - a) \cdot f_A(a) \;\leqslant\; \sum_{a=0}^{\infty} f_A(a) \;=\; 1\,,$$

the inequality above holds since $f(\sigma) \leqslant 1$ by definition. ■

Next, we point out another simple fact; for values of σ that $f(\sigma) = 1$ or $f(\sigma + \rho) = 0$, condition (3) is always satisfied. This is easy to show by using Lemma 3: For σ that $f(\sigma) = 1$, we have

$$\begin{aligned}&(f * f_A)(\sigma + \rho) \cdot f(\sigma) \;+\; f(\sigma + \rho) \cdot F(\sigma) \\ &\quad = (f * f_A)(\sigma{+}\rho) \cdot 1 \;+\; f(\sigma{+}\rho) \cdot 0 \;=\; (f * f_A)(\sigma{+}\rho) \\ &\quad \leqslant 1 \;=\; f(\sigma)\,,\end{aligned} \tag{12}$$

where the inequality follows by Lemma 3. Similarly, for σ that $f(\sigma + \rho) = 0$, we have

$$\begin{aligned}&(f * f_A)(\sigma + \rho) \cdot f(\sigma) \;+\; f(\sigma + \rho) \cdot F(\sigma) \\ &= (f * f_A)(\sigma{+}\rho) {\cdot} f(\sigma) \;+\; 0 \cdot F(\sigma) = (f * f_A)(\sigma{+}\rho) {\cdot} f(\sigma) \\ &\leqslant f(\sigma)\,,\end{aligned} \tag{13}$$

where the inequality follows again by Lemma 3.

Thus, it is necessary and sufficient that we show condition (3) is also satisfied for any possible solution f_A over the following set Σ defined as

$$\Sigma \triangleq \big\{\sigma :\; 0 < f(\sigma + \rho),\; f(\sigma) < 1\big\}. \tag{14}$$

A solution for f_A is found by the following three steps.

Step 1) If Σ is empty, then it follows by (12) and (13) that any valid pmf provides a solution for f_A satisfying condition (3). However, there exists a selection of f_As with which Q-Composer generates flows that make statement (4) be satisfied with equality for any positive n—such an equality is desired for tightness of performance bounds. Below, we choose a simple such f_A.

Let D denote the set $\{\sigma :\; f(\sigma) > 0\}$. The maximum of D exists, which we prove next: It suffices to show that D is not empty and that any σ in D is less than a constant. For any negative σ, $f(\sigma) = 1$ which holds by the 2nd property assumed for any bounding function (see Sec. 2). Thus, D is not empty. Secondly, the emptiness of Σ implies that for any σ either $f(\sigma) = 1$ or $f(\sigma + \rho) = 0$. For all σ, $f(\sigma)$ can not be equal to 1; otherwise, the 3rd property assumed for any bounding function (see Sec. 2) is violated. So, there must exist a σ^* for which $f(\sigma^* + \rho) = 0$. Since f is nonincreasing, any σ in D must be less than $\sigma^* + \rho$. This concludes the proof.

Let σ_1 denote the maximum of D;

$$\sigma_1 \triangleq \max\{\sigma :\; f(\sigma) > 0\}\,.$$

For simplicity, we choose

$$f_A(a) := \begin{cases} 1 & \text{if } a = \sigma_1 + \rho, \\ 0 & \text{else,} \end{cases} \tag{15}$$

and stop.

If Σ is not empty, then go to Step 2.

Step 2) Σ is not empty. The minimum of Σ exists, which is easy to show: It suffices to show that any σ in Σ

is greater than a constant. Since $f(\sigma)=1$ for any $\sigma<0$, which holds by the 2nd property assumed for any bounding function (see Sec. 2), any σ in Σ has to be positive.

Let σ_0 denote the minimum of Σ;

$$\sigma_0 \triangleq \min \Sigma.$$

We point out two simple facts about Σ by the following two lemmas.

Lemma 4 *For any σ in Σ, $f(\sigma)>0$.*

Proof: For any σ in Σ, we have

$$f(\sigma) \geqslant f(\sigma+\rho) > 0\,;$$

the first inequality holds since ρ is nonnegative and f is nonincreasing, the second inequality holds since $\sigma \in \Sigma$. ■

Lemma 5 *For any σ less than σ_0, $f(\sigma)=1$.*

Proof: Any σ less than σ_0 is not in Σ by the definition of σ_0. Hence for any σ less than σ_0, either $f(\sigma)=1$ or $f(\sigma+\rho)=0$, which follows from the definition of Σ. The later statement (i.e. $f(\sigma+\rho)=0$) can not be true since

$$f(\sigma+\rho) \geqslant f(\sigma_0+\rho) > 0\,;$$

the first inequality holds since $\sigma<\sigma_0$ and f is nonincreasing, the second inequality holds since σ_0 belongs to Σ. Therefore, $f(\sigma)$ must be equal to 1. ■

Now for $\sigma=\sigma_0$, the inequality in condition (3), which needs to be satisfied, can be manipulated as follows;

$$\begin{aligned}(f*f_A)(\sigma_0+\rho)\cdot f(\sigma_0) \ + \ f(\sigma_0+\rho)\cdot F(\sigma_0) &\leqslant f(\sigma_0)\\ (f*f_A)(\sigma_0+\rho) &\leqslant \frac{f(\sigma_0)-f(\sigma_0+\rho)\cdot F(\sigma_0)}{f(\sigma_0)}\\ &= 1-\frac{f(\sigma_0+\rho)\cdot F(\sigma_0)}{f(\sigma_0)}\,, \qquad (16)\end{aligned}$$

the division above is allowed since $f(\sigma_0)>0$ by Lemma 4.

The convolution $(f*f_A)(\sigma_0+\rho)$ above can be manipulated as follows;

$$\begin{aligned}(f*f_A)(\sigma_0+\rho) &= \sum_{a=0}^{\infty} f(\sigma_0+\rho-a)\cdot f_A(a)\\ &= \sum_{a=0}^{\rho} f(\sigma_0+\rho-a)\cdot f_A(a) + \sum_{a>\rho} f(\sigma_0+\rho-a)\cdot f_A(a),\end{aligned}$$

since $f(\sigma)=1$ for any $\sigma<\sigma_0$ by Lemma 5, we get

$$\begin{aligned}&= \sum_{a=0}^{\rho} f(\sigma_0+\rho-a)\cdot f_A(a) + \sum_{a>\rho} f_A(a)\\ &= \sum_{a=0}^{\rho} f(\sigma_0+\rho-a)\cdot f_A(a) + \big(1-F_A(\rho)\big).\end{aligned}$$

Substituting the last equality above into (16) and flipping $1-F_A(\rho)$ over to the other side, we get

$$\sum_{a=0}^{\rho} f(\sigma_0+\rho-a)\cdot f_A(a) \leqslant F_A(\rho) - \frac{f(\sigma_0+\rho)\cdot F(\sigma_0)}{f(\sigma_0)}. \qquad (17)$$

The rhs of (17) must be nonnegative as the lhs is. In other words, $F_A(\rho)$ has to satisfy

$$F_A(\rho) \geqslant \frac{f(\sigma_0+\rho)\cdot F(\sigma_0)}{f(\sigma_0)}\,. \qquad (18)$$

This further implies that $F_A(\rho)$ has to be positive as the term on the rhs in (18) is, which holds since $f(\sigma_0+\rho)>0$ and $f(\sigma_0)<1$ as σ_0 belongs to Σ. This means that for at least one a in $[0,\rho]$, $f_A(a)$ has to be positive.

At this point we introduce another condition that we would like f_A to satisfy in addition to condition (3):

The value of $f_A(a)$ being found by the current method is progressively as small as possible for any a in $\{a:\ a\leqslant \rho+\sigma-\sigma_0,\ \sigma\in\Sigma\}$. (19)

By 'progressively' in condition (19) we mean that for any a_1 less than a_2, where a_1 and a_2 belong to the set specified in (19), $f_A(a_2)$ is as small as possible subject to $f_A(a_1)$ being as small as possible in the current method of finding f_A. In other words, we first make sure that $f_A(a_1)$ is as small as possible, then make sure that $f_A(a_2)$ is as small as possible. How the set in (19) is specified will be clear later on by the beginning of Step 3; for the time being one can consider it as just a set of a's.

We would want condition (19) be satisfied in addition to (3), for the purpose of having tight performance bounds.

Now, since we know that for at least one a in $[0,\rho]$ $f_A(a)$ has to be positive, let us choose $f_A(a):=0$ for any $a<\rho$; as motivated by condition (19). However, to be able to make these assignments, we need to check if (17) can be satisfied with this choice of $f_A(a)$'s. Substituting these values of f_A into (17), we get;

$$\begin{aligned}f(\sigma_0)\cdot f_A(\rho) &\leqslant f_A(\rho) - \frac{f(\sigma_0+\rho)\cdot F(\sigma_0)}{f(\sigma_0)}\\ f_A(\rho) - f(\sigma_0)\cdot f_A(\rho) &\geqslant \frac{f(\sigma_0+\rho)\cdot F(\sigma_0)}{f(\sigma_0)}\\ f_A(\rho)\cdot\big(1-f(\sigma_0)\big) &\geqslant \frac{f(\sigma_0+\rho)\cdot F(\sigma_0)}{f(\sigma_0)}\\ f_A(\rho)\cdot F(\sigma_0) &\geqslant \frac{f(\sigma_0+\rho)\cdot F(\sigma_0)}{f(\sigma_0)}\\ f_A(\rho) &\geqslant \frac{f(\sigma_0+\rho)}{f(\sigma_0)}\,,\end{aligned}$$

in obtaining the last inequality note that $F(\sigma_0)>0$ since $f(\sigma_0)<1$ as σ_0 belongs to Σ. To satisfy condition (19), we choose $f_A(\rho)$ to be equal to the rhs of the last inequality above. Thus, in Step 2, we make the following assignments;

$$f_A(a) := \begin{cases} 0 & \text{if } \ a<\rho,\\ \frac{f(\sigma_0+\rho)}{f(\sigma_0)} & \text{if } \ a=\rho.\end{cases} \qquad (20)$$

Note that the assignments in (20) satisfy (18).

Go to Step 3.

Step 3) Suppose that for some σ in Σ, where $\sigma \geqslant \sigma_0$, we have found an assignment for $f_A(a)$ for all a less than or equal to $\rho + \sigma - \sigma_0$ by the solution that we are presenting (i.e. the current assignments make condition (3) be satisfied for all σ less than or equal to the σ being supposed). This statement clearly holds by Step 2. Let us represent the σ being supposed as $\sigma = \sigma_0 + k - 1$, where $k \geqslant 1$. Note that the largest value of a that an assignment has been made so far in the current assignment of f_A (i.e. $\rho + \sigma - \sigma_0$ by the supposition) is equal to $\rho + k - 1$.

If $\sigma + 1$ is not in Σ, then we stop and make the assignment $f_A(\rho + k) := 1 - F_A(\rho + k - 1)$. Else, we proceed as follows.

It is given that $\sigma+1$ belongs to Σ. We find an assignment for $f_A(\rho + k)$ by considering the inequality corresponding to $\sigma + 1$ in condition (3) (i.e. the inequality obtained by replacing σ by $\sigma + 1$ in condition (3)). This inequality needs to be satisfied. We manipulate this inequality as we have manipulated the one in obtaining (16), and get

$$(f * f_A)(\sigma + 1 + \rho) \leqslant 1 - \frac{f(\sigma + 1 + \rho) \cdot F(\sigma + 1)}{f(\sigma + 1)} \,. \qquad (21)$$

Note that in getting (21), the division by $f(\sigma+1)$ is allowed since $f(\sigma + 1) > 0$ by Lemma 4 (as $\sigma + 1$ belongs to Σ).

The convolution $(f * f_A)(\sigma + 1 + \rho)$ can be manipulated as follows;

$$\begin{aligned}
&(f * f_A)(\sigma + 1 + \rho) = (f * f_A)(\sigma_0 + k + \rho) \\
&= \sum_{a=0}^{\infty} f(\sigma_0 + k + \rho - a) \cdot f_A(a) \\
&= \sum_{a=0}^{\rho+k} f(\sigma_0+k+\rho-a)\cdot f_A(a) + \sum_{a>\rho+k} f(\sigma_0+k+\rho-a)\cdot f_A(a),
\end{aligned}$$

since $f(u) = 1$ for any $u < \sigma_0$ by Lemma 5, we get

$$\begin{aligned}
&= \sum_{a=0}^{\rho+k} f(\sigma_0 + k + \rho - a) \cdot f_A(a) + \sum_{a>\rho+k} f_A(a) \\
&= \sum_{a=0}^{\rho+k} f(\sigma_0 + k + \rho - a) \cdot f_A(a) + \big(1 - F_A(\rho + k)\big).
\end{aligned}$$

Substituting the last equality above into (21), flipping $1 - F_A(\rho + k)$ over to the other side, and replacing $\sigma + 1$ by $\sigma_0 + k$, we get

$$\sum_{a=0}^{\rho+k} f(\sigma_0 + k + \rho - a) \cdot f_A(a) \leqslant F_A(\rho + k) - \frac{f(\sigma_0 + k + \rho) \cdot F(\sigma_0 + k)}{f(\sigma_0 + k)} \,. \qquad (22)$$

Note that the rhs of (22) must be nonnegative as the lhs is; i.e. $F_A(\rho + k)$ has to satisfy

$$F_A(\rho + k) \geqslant \frac{f(\sigma_0 + k + \rho) \cdot F(\sigma_0 + k)}{f(\sigma_0 + k)} \,. \qquad (23)$$

We manipulate (22) by pulling up the term corresponding to $a = \rho + k$ in the summation and the term $f_A(\rho + k)$ in $F_A(\rho + k)$ to the lhs of the inequality, and putting all the rest of the terms to the other side. With that, we obtain (25)—see the next page.

We would consider assigning the quantity on the rhs of (25) to $f_A(\rho+k)$, as motivated by condition (19). But, we first need to compare this quantity with $1 - F_A(\rho + k - 1)$ corresponding to the current assignments of f_A to ensure that $F_A(\rho + k) \leqslant 1$ in order to have a valid pmf.

If this quantity being mentioned above is less than or equal to $1 - F_A(\rho + k - 1)$, then we take the maximum of this quantity with 0, assign that maximum to $f_A(\rho + k)$, increment k by 1, and go to the beginning of Step 3.

<u>Else</u> (i.e. the rhs of (25) is greater than $1 - F_A(\rho + k - 1)$ corresponding to the current assignments of f_A), we need to increase the value of $f_A(a)$ for some a's less than or equal to $\rho + k - 1$ in the current assignment so that we can have the rhs of (25) as less than or equal to $1 - F_A(\rho + k - 1)$ corresponding to the new assignment. This is made possible in this case (i.e. the '<u>Else</u>' case), since the fraction multiplying $f_A(a)$ in the summation in (25) is greater than 1 for at least one a in $[0, \rho + k - 1]$; this is what we show next.

For the rhs of (25) to be greater than $1 - F_A(\rho + k - 1)$ for any assignment of f_A, the fraction $\frac{F(\sigma_0+k)}{F(\sigma_0)}$ must be greater than 1: Note that $\frac{F(\sigma_0+k)}{F(\sigma_0)} \geqslant 1$ since F is nondecreasing and $k \geqslant 1$. Suppose that this fraction is equal to 1, then we have

the rhs of (25)

$$= \frac{f(\sigma_0 + k + \rho)}{f(\sigma_0 + k)} - \sum_{a=0}^{\rho+k-1} f_A(a) \cdot \frac{F(\sigma_0 + k + \rho - a)}{F(\sigma_0)},$$

since f is nonincreasing and ρ is nonnegative, we get

$$\leqslant 1 - \sum_{a=0}^{\rho+k-1} f_A(a) \cdot \frac{F(\sigma_0 + k + \rho - a)}{F(\sigma_0)},$$

similarly, as F is nondecreasing, the fraction $\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)}$ for any $a \leqslant \rho+k-1$ is greater than or equal to 1, hence

$$\leqslant 1 - \sum_{a=0}^{\rho+k-1} f_A(a) \;=\; 1 - F_A(\rho + k - 1)\,.$$

This concludes proving the claim that $\frac{F(\sigma_0+k)}{F(\sigma_0)} > 1$.

We try to find a new assignment for f_A by increasing the value of only one $f_A(a)$ for an $a \leqslant \rho + k - 1$ in the current assignment of f_A and by *temporarily* nullifying the value of $f_A(a)$ for all a greater than the a that we choose

$$f(\sigma_0)\cdot f_A(\rho+k) - f_A(\rho+k) \leqslant -\sum_{a=0}^{\rho+k-1} f(\sigma_0+k+\rho-a)\cdot f_A(a) + \sum_{a=0}^{\rho+k-1} f_A(a) - \frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)}$$

$$f_A(\rho+k)\cdot\big(1-f(\sigma_0)\big) \geqslant \frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)} - \sum_{a=0}^{\rho+k-1} f_A(a)\cdot\big(1-f(\sigma_0+k+\rho-a)\big)$$

$$f_A(\rho+k)\cdot F(\sigma_0) \geqslant \frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)} - \sum_{a=0}^{\rho+k-1} f_A(a)\cdot F(\sigma_0+k+\rho-a) \tag{24}$$

$$f_A(\rho+k) \geqslant \frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - \sum_{a=0}^{\rho+k-1} f_A(a)\cdot \frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)}\,, \tag{25}$$

the division in obtaining the last line above is allowed since $f(\sigma_0) < 1$ as σ_0 belongs to Σ.

to increase the value of $f_A(a)$. Let a that we choose to increase the value of $f_A(a)$ be represented as $\rho+k-k^*$ for some $k^* \geqslant 1$. To determine a new assignment for f_A, we need to consider satisfying inequality (26) corresponding to the new assignment—see the next page. With that, we obtain (28).

Motivated by condition (19), we would want to update the value of $f_A(\rho+k-k^*)$ by the quantity on the lhs of (28). However, we first need to make sure that such an assignment is feasible; i.e. this quantity should be i) nonnegative, ii) less than or equal to 1, iii) allowing a valid probability assignment, and iv) greater than or equal to the previously assigned value of $f_A(\rho+k-k^*)$.

The restrictions 'i)' and 'iv)' hold by the '<u>Else</u>' case, which are shown by (29)—see the next page. Since the previous assignment of f_A is a valid assignment, (29) shows that the quantity that we are hoping to update the value of f_A at $\rho+k-k^*$ is positive and greater than the previously assigned value of $f_A(\rho+k-k^*)$. We need restriction 'iv)' to hold in order for condition (3) to remain satisfied for σ being equal to σ_0+k-k^*—this σ is not the σ being supposed at the beginning of Step 3.

For the other two restrictions, i.e. 'ii)' and 'iii)' in the paragraph before the previous one, we need to show that the lhs of (28) is less than or equal to $1-F_A(\rho+k-k^*-1)$. This may or may not be true, thus we choose a k^* as defined in (30)—see that last page. The minimum in the definition of k^* is motivated by condition (19).

We need to check if such a minimum defined in (30) does always exist or not. First, let us inspect if in this case (i.e. the '<u>Else</u>' case) the set over which the minimum is take in (30) is always nonempty. For $u=k$, the restriction on u that $\frac{F(\sigma_0+u)}{F(\sigma_0)} > 1$ always holds as we already know that $\frac{F(\sigma_0+k)}{F(\sigma_0)} > 1$. The other restriction on $u=k$ also holds as shown by (31)—see the last page—; note that (31) corresponds to the other restriction on $u=k$. So, the set over which the minimum is take in (30) is always nonempty. Secondly, the minimum of the set in (30) always exist since any u in the set has to be greater than or equal to 1 by the restriction that $\frac{F(\sigma_0+u)}{F(\sigma_0)} > 1$.

Thus, as we now know that the quantity on the lhs of (28) satisfies all the restrictions 'i)' through 'iv)' stated four paragraphs ago, we can update the value of $f_A(\rho+k-k^*)$ by this quantity. We do this update. Furthermore, if $k^*=1$, then we assign $f_A(\rho+k) := 1-F_A(\rho+k-k^*)$ and increment k by 1. (This assignment of f_A follows by backtracking (28) with equality all the way up to (26) and then comparing (26) with (25).) Else (i.e. $k^*>1$), we set $\sigma := \sigma_0+k-k^*$. Go to the beginning of Step 3.

Finally, we note that constraint (23), which is relevant only for the assignments to be made for the $k^*=1$ case or before the '<u>Else</u>' case, is satisfied by these assignments; this can be noted by (24). This ends Step 3.

We summarize the above solution for f_A by giving the pseudocode for Composer-pmf (i.e. the algorithm finding this solution).

Composer-pmf

<u>Input:</u> A rate ρ and a bounding function f.

<u>Output:</u> A pmf f_A which satisfies condition (3).

<u>Body:</u> Determine the set Σ defined in (14),
  if Σ is empty, then
    determine σ_1 (i.e. the maximum of $\{\sigma : f(\sigma) > 0\}$),
    $f_A(\sigma_1+\rho) := 1$,
    $f_A(a) := 0$ for all $a \neq \sigma_1+\rho$,
  else,
    determine σ_0 (i.e. the minimum of Σ),
    make the assignments in (20),
    $k := 1$,
    while σ_0+k belongs to Σ, do
      compute the quantity on the rhs in (25),
      call this quantity `rhs-of-25`,
      if `rhs-of-25` $\leqslant 1-F_A(\rho+k-1)$, then
        $f_A(\rho+k) := \max\{0, \texttt{rhs-of-25}\}$,
        $k := k+1$,
      else,
        determine k^* defined in (30),
        compute the quantity on the lhs in (28),

$$\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - \sum_{a=0}^{\rho+k-1} f_A(a)\cdot\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} \leqslant 1 - F_A(\rho+k-1), \tag{26}$$

since we temporarily choose for the new assignment that $f_A(a) := 0$ for any a in $(\rho+k-k^*, \rho+k-1]$, we have

$$\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - \sum_{a=0}^{\rho+k-k^*} f_A(a)\cdot\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} \leqslant 1 - F_A(\rho+k-k^*), \tag{27}$$

by pulling up the term corresponding to $a=\rho+k-k^*$ in the summation above and the term $f_A(\rho+k-k^*)$ in $F_A(\rho+k-k^*)$ to the rhs, and putting all the rest of the terms to the other side, we get

$$\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - 1 - \sum_{a=0}^{\rho+k-k^*-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} - 1\right] \leqslant f_A(\rho+k-k^*)\cdot\left[\frac{F(\sigma_0+k^*)}{F(\sigma_0)} - 1\right],$$

now if we choose k^* such that $\frac{F(\sigma_0+k^*)}{F(\sigma_0)} > 1$, we get

$$\frac{\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - 1 - \sum_{a=0}^{\rho+k-k^*-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} - 1\right]}{\frac{F(\sigma_0+k^*)}{F(\sigma_0)} - 1} \leqslant f_A(\rho+k-k^*)\,. \tag{28}$$

For the current assignment of f_A, we have by the '<u>Else</u>' case that

$$\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - \sum_{a=0}^{\rho+k-1} f_A(a)\cdot\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} > 1 - F_A(\rho+k-1),$$

by pulling up all the terms corresponding to a's, where $\rho+k-k^* < a \leqslant \rho+k-1$, in the summation on the lhs above to the rhs of the inequality, and separating the terms in $F_A(\rho+k-k^*)$ corresponding to the same a's, we get

$$\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - \sum_{a=0}^{\rho+k-k^*} f_A(a)\cdot\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} > 1 - \sum_{a=0}^{\rho+k-k^*} f_A(a) + \sum_{a=\rho+k-k^*+1}^{\rho+k-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} - 1\right],$$

since $\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} \geqslant 1$ for any a in $[0, \rho+k-1]$, the last inequality above implies that we also have

$$\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - \sum_{a=0}^{\rho+k-k^*} f_A(a)\cdot\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} > 1 - \sum_{a=0}^{\rho+k-k^*} f_A(a),$$

applying the manipulations performed for the set of inequalities from (27) to (28) to the above inequality, we get

$$\frac{\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - 1 - \sum_{a=0}^{\rho+k-k^*-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} - 1\right]}{\frac{F(\sigma_0+k^*)}{F(\sigma_0)} - 1} > f_A(\rho+k-k^*)\,, \tag{29}$$

where $f_A(\rho+k-k^*)$ above is the previously assigned value to f_A at $\rho+k-k^*$.

$$k^* \triangleq \min\left\{ u:\ u \leqslant k,\ \frac{F(\sigma_0+u)}{F(\sigma_0)} > 1,\ \frac{\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - 1 - \sum_{a=0}^{\rho+k-u-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} - 1\right]}{\frac{F(\sigma_0+u)}{F(\sigma_0)} - 1} \leqslant 1 - \sum_{a=0}^{\rho+k-u-1} f_A(a) \right\}. \quad (30)$$

$$\sum_{a=0}^{\rho-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} - 1\right] \geqslant \sum_{a=0}^{\rho-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k)}{F(\sigma_0)} - 1\right]$$

$$\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - 1 - \sum_{a=0}^{\rho-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} - 1\right] \leqslant \frac{F(\sigma_0+k)}{F(\sigma_0)} - 1 - \sum_{a=0}^{\rho-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k)}{F(\sigma_0)} - 1\right]$$

$$\frac{\frac{f(\sigma_0+k+\rho)\cdot F(\sigma_0+k)}{f(\sigma_0+k)\cdot F(\sigma_0)} - 1 - \sum_{a=0}^{\rho-1} f_A(a)\cdot\left[\frac{F(\sigma_0+k+\rho-a)}{F(\sigma_0)} - 1\right]}{\frac{F(\sigma_0+k)}{F(\sigma_0)} - 1} \leqslant 1 - \sum_{a=0}^{\rho-1} f_A(a)\,, \quad (31)$$

the first inequality above holds since the smallest fraction multiplying $f_A(a)$ on the lhs is equal to $\frac{F(\sigma_0+k+1)}{F(\sigma_0)} - 1$ which is greater than or equal to the identical fractions $\frac{F(\sigma_0+k)}{F(\sigma_0)} - 1$ on the rhs; the second inequality holds since $\frac{f(\sigma_0+k+\rho)}{f(\sigma_0+k)} \leqslant 1$ as f is nonincreasing and ρ is nonnegative; finally, the division in obtaining the last inequality is allowed since we already know that $\frac{F(\sigma_0+k)}{F(\sigma_0)} > 1$.

call this quantity `lhs-of-28`,
$f_A(\rho+k-k^*) :=$ `lhs-of-28`,
if $k^* = 1$, then
$f_A(\rho+k) := 1 - F_A(\rho+k-k^*)$,
$k := k+1$,
else, $k := k - k^* + 1$
end if,
end if,
end while,
end if,
$f_A(\rho+k) := 1 - F_A(\rho+k-1)$.
End Composer-pmf.

C Proof of Theorem 2

Let T denote an integer time specified as Input to CpR. Feed any traffic obtained by CpR for the given Input into a work-conserving server with rate ρ, where the time origin is at T. Let Q_r denote the queue-size of the server.

First, we prove that the following simple statement holds;

$$Q_r(T+n-1) \leqslant Q(n) \quad \text{for all nonnegative } n, \quad (32)$$

where $Q(n)$ is given by (5). The proof follows by mathematical induction on n: (*Basis*) $Q_r(T-1) = 0 = Q(0)$ which holds by the convention in this text that any work-conserving server is initially empty, that the time origin is at T, and $Q(0) = 0$ as specified in CpR. (*Induction Step*) Suppose that (32) holds for some nonnegative n—this is true by the induction *Basis* which corresponds to $n = 0$. The following relations show that (32) holds also for $n+1$;

$$\begin{aligned} Q_r(T+n) &= \max\{0,\, Q_r(T+n-1) + r(T+n) - \rho\} \\ &\leqslant \max\{0,\, Q(n) + r(T+n) - \rho\} \\ &\leqslant \max\{0,\, Q(n) + r_{\text{syn}}(n+1) - \rho\} \\ &= Q(n+1), \end{aligned}$$

where the first line follows by how a work-conserving server is defined to work, the second line follows by the induction hypothesis, the third line follows by (6). This concludes showing that (32) holds.

The following relations hold for any $n \geqslant 0$ and all σ;

$$\{Q_r(T+n-1) > \sigma\} \subseteq \{Q(n) > \sigma\}$$

taking the probabilities of both sides, we get

$$\begin{aligned} \mathrm{P}(Q_r(T+n-1) > \sigma) &\leqslant \mathrm{P}(Q(n) > \sigma) \\ &\leqslant f(\sigma), \end{aligned}$$

where the subset relation follows by (32) and the last line follows by Theorem 1.

For any $n < 0$ and for all σ, the following inequality

$$\mathrm{P}(Q_r(T+n-1) > \sigma) \leqslant f(\sigma) \quad (33)$$

also holds, since $Q_r(T+n-1) = 0$ (which holds by the convention in this text that any work-conserving server is initially empty and that the time origin is at T) and by the 2nd property assumed for any bounding function. ∎