Convexity and Optimal Load Distributions in Work Conserving */*/1 Queues

Michael J. Neely
MIT -- LIDS
mjneely@mit.edu

Eytan Modiano
MIT -- LIDS
modiano@mit.edu

***Abstract* -- In this paper we develop fundamental convexity properties of unfinished work and packet waiting time in a work conserving */*/1 queue. The queue input consists of an uncontrollable background process and a rate-controllable input stream. We show that any moment of unfinished work is a convex function of the controllable input rate. The convexity properties are then extended to address the problem of optimal routing of arbitrary input streams over a collection of N queues in parallel with different (possibly time-varying) linespeeds ($\mu_1(t),...,\mu_N(t)$). Our convexity results hold for stream-based routing (where individual packet streams must be routed to the same queue) as well as for packet-based routing where each packet is routed to a queue using some pre-determined splitting method, such as probabilistic splitting. Our analysis of these general systems is carried out by introducing a new function of the superposition of two input streams that we call the *blocking function*. Using this function facilitates analysis and provides much insight into the sample path dynamics of */*/1 queues.**

I. Introduction

In this paper we examine a work conserving */*/1 queue and develop fundamental monotonicity and convexity properties of unfinished work and packet waiting time in the queue as a function of the packet arrival rate λ. The "*/*" notation refers to the fact that the input process has arbitrarily distributed and correlated interarrival times and packet lengths. (This differs from the standard GI/GI description, where interarrival times and packet lengths are independent and identically distributed). The arrival process consists of the superposition of two component streams: an arbitrary and uncontrollable background input of the */* type, and a rate-controllable packet stream input (Fig. 1). The rate-controllable stream contains a collection of indistinguishable */* substreams, and its rate is varied in discrete steps by adding or removing these substreams as inputs to the queue. We show that any moment of unfinished work is a convex function of this input rate. Under the special case of FIFO service, we show that waiting time moments are also convex.

We then extend the convexity result to address the problem of optimally routing input streams over a parallel collection of N queues with different linespeeds ($\mu_1,...,\mu_N$). We show that cost functions consisting of convex combinations of unfinished work moments in each of the queues are convex in the N-dimensional rate tuple ($\lambda_1,...,\lambda_N$). In the symmetric case where the N queues are weighted equally in the cost function and have identical background processes, this convexity result implies that the uniform rate allocation minimizes cost. In the

† This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the Next Generation Internet (NGI) initiative.

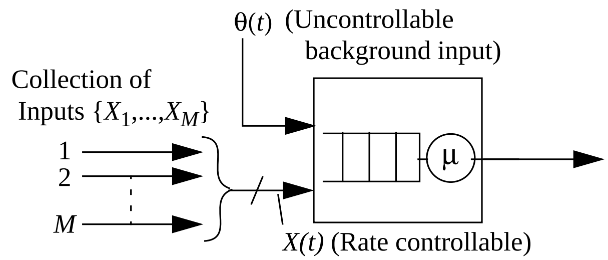

Figure 1: A work conserving queue with server linespeed μ, a */* background input $\theta(t)$, and rate-controllable */* inputs $X(t) = \{X_1(t),...,X_M(t)\}$.

case of an asymmetric collection of N parallel queues, we develop a sequentially greedy routing algorithm that is optimal.

The convexity results and optimization methods are extended to treat queues with time-varying linespeeds ($\mu_1(t),..., \mu_N(t)$). We show that the amount of unprocessed bits in the multi-queue system remains convex in the input rate vector ($\lambda_1,..., \lambda_N$). However, we demonstrate that waiting times are not necessarily convex for general time varying linespeed problems. For simplicity of exposition, we postpone the time-varying analysis until section VI.

Convexity of single and parallel collections of queues has been addressed previously with various assumptions about the nature of the input processes and the service time processes. In [1], the authors develop a convexity theory of "multi-modular functions" and use this theory to develop an optimal admission control in a D/D/1 queue with fixed batch arrivals. In [2,3], the authors analyze the expected packet occupancy in tree networks of deterministic service time queues. It is shown that expected occupancy of any interior queue of the tree is a concave function of the multiple exogenous input rates, while occupancy in queues on the edge of the network are shown to be convex. Convexity properties of parallel GI/GI/1 queues with packet-based probabilistic "Bernoulli" routing are developed in [4].

Our treatment of the convexity problem for streams of inputs is an important feature, since packets from a single source often must be routed together to maintain predictability and to prevent out-of-order delivery. However, we also treat the packet-based routing method of [4] in a more general (yet simpler) context. Rather than emphasizing the differences between packet-based and stream-based routing, we discover a fundamental similarity. We consider packet-based routing of a general */* input stream whose rate can be split according to a continuous rate parameter, using a splitting method such as the probabilistic "Bernoulli" splitting in [4]. We find that convexity for this packet-based routing problem is a consequence of our stream-based results.

Our analysis is carried out by introducing a new function of the superposition of two input streams that we call the *blocking function*. Properties of the blocking function are developed by examining sample paths of work conserving queues, and each property corresponds to an intuitive comparison of two different queueing system configurations. These properties are then used to establish the convexity and optimal routing results.

II. The Blocking Function For */*/1 Queues

Consider a work conserving queue with a single server that can process packets at a line speed of μ bits/second (Fig. 2). Variable length packets from input stream X flow into the queue and are processed at the single server according to any work-conserving service discipline (e.g., FIFO, LIFO, Shortest Packet First, GPS, etc.). The input stream is characterized according to two random processes: (*i*) The sequence $\{a_k\}$ of inter-arrival times, and (*ii*) The sequence $\{l_k\}$ of packet lengths.

We assume the processes $\{a_k\}$ and $\{l_k\}$ are ergodic with arrival rate λ and average packet length $E(L)$, respectively. In general, inter-arrival times may be correlated with each other as well as jointly correlated with the packet length process. We maintain this generality by describing the input to the queue by the single random process $X(t)$, which represents the amount of bits brought into the queue as a function of time. As shown in Fig. 2, a particular input $X(t)$ is a non-decreasing staircase function. Jumps in the $X(t)$ function occur at packet arrival epochs, and the amount of increase at these times is equal to the length of the entering packet. The accumulated amount of *work* brought into the queue can be written as $X(t)/\mu$, which has units of time.

For a given queue with input process $X(t)$, we represent the amount of *unfinished work* in the system at time t as $U_X(t)$--the total amount of time for all packets in the queueing system (queue plus server) to empty if no more packets were to arrive. We assume the queue is initially empty at time t=0. It is clear that $U_X(t)$ is the same for all work conserving service disciplines. It is completely determined by $X(t)$ as well as the server linespeed μ. An example unfinished work function $U_X(t)$ is shown in Fig. 2. Notice the triangular structure and the fact that each new triangle emerges at packet arrival times and has a downward slope of -1.

Now consider a new input stream X_1+X_2 which is the superposition of two input streams X_1, X_2. We make the following sample path observation, which holds for any arbitrary set of sample paths $X_1(t)$, $X_2(t)$:

<u>*Observation 1*</u>*:* For all times t, we have:

$$U_{X1+X2}(t) \geq U_{X1}(t) + U_{X2}(t) \quad . \qquad (1)$$

Thus, for any two inputs X_1 and X_2, the amount of unfinished work in a work conserving queueing system with the superposition process X_1+X_2 is *always* greater than or equal to the sum of the work in two identical queues with these same processes X_1 and X_2 entering them individually. This is illustrated in Fig. 3.

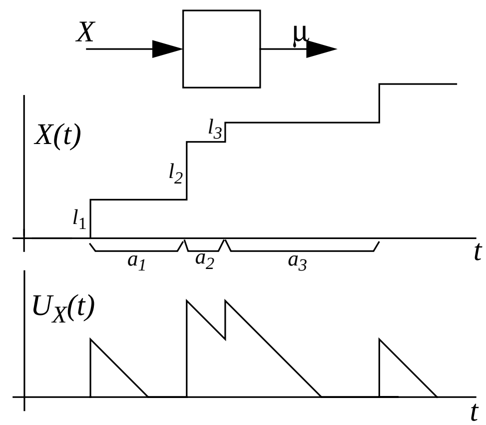

Figure 2: A work conserving */*/1 queue, and typical sample paths of accumulated and unfinished work.

<u>*Proof of Observation 1*</u>*:* We compare the two system configurations of Fig. 3. Since $U_{X1+X2}(t)$ is the same for all work conserving service disciplines, we can imagine that packets from the X_1 stream have preemptive priority over X_2 packets. The queueing dynamics of the X_1 packets are therefore unaffected by any low priority X_2 packets. Thus, the $U_{X1+X2}(t)$ function can be written as $U_{X1}(t)$ plus an extra amount $extra\_X_2(t)$ due to the X_2 packets, as shown in Fig. 4. This extra amount (represented as the striped region in Fig. 4) can be thought of as the amount of unfinished work remaining in the queue with the X_2 input stream alone, where the server goes on idle "vacations" exactly at times when $U_{X1}(t)$ is nonzero. Clearly, this unfinished work is greater than or equal to the unfinished work there would be if the server did not go on vacations--which is $U_{X2}(t)$. Thus:

$$U_{X1+X2}(t) = U_{X1}(t) + extraX_2(t) \geq U_{X1}(t) + U_{X2}(t) \text{ . } \square$$

This simple observation motivates the following definition:

Definition: The *Blocking Function* $\beta_{X1,\ X2}(t)$ between two streams X_1 and X_2 is the function:

$$\beta_{X1,X2}(t) = U_{X1+X2}(t) - U_{X1}(t) - U_{X2}(t) . \qquad (2)$$

Thus, the blocking function is a random process which represents the extra amount of unfinished work in the system due to the blocking incurred by packets from the X_1 stream mixing with the X_2 stream. From this definition, we immediately find for all times t:

<u>*Lemma 1*</u>*:* $\beta_{X1,X2}(t) \geq 0$ (non-negativity) (3)
<u>*Lemma 2*</u>*:* $\beta_{X1,X2}(t) = \beta_{X2,X1}(t)$ (symmetry) (4)
<u>*Lemma 3*</u>: $\beta_{X1+X2,X3}(t) \geq \beta_{X1,X3}(t)$ (monotonicity) (5)

The non-negativity lemma above is just a re-statement of (1), while the symmetry property is obvious from the blocking

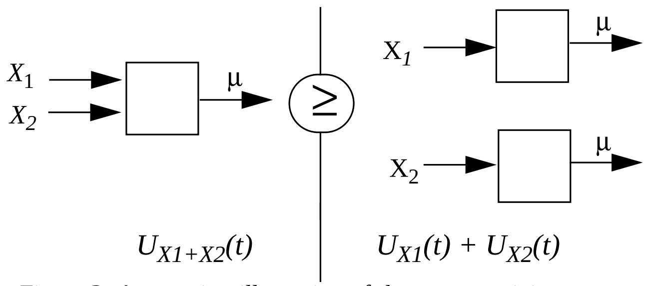

Figure 3: A queueing illustration of the non-negativity property of the blocking function.

function definition. Below we prove the monotonicity lemma.

Proof of Lemma 3 (Monotonicity): From the definition of the blocking function in (2), we find that the monotonicity statement is equivalent to the following inequality at every time t:

$$U_{X1+X2+X3}(t) + U_{X1}(t) \geq U_{X1+X2}(t) + U_{X1+X3}(t) \quad (6)$$

We have illustrated (6) in Fig. 5. We thus prove that the sum of the unfinished work in Systems A and B of Fig. 5 is greater than or equal to the sum in A' and B'.

In a manner similar to the proof of Observation 1, we give packets from both the X_1 and X_2 streams preemptive priority over X_3 packets. The queues of Fig. 5 can thus be treated as having servers that take "vacations" from serving X_3 packets during busy periods caused by the other streams. Comparing the A and A' systems, as well as the B and B' systems, we have:

$$U_{X1+X2+X3}(t) = U_{X1+X2}(t) + extra\_in\_System\_A(t) \quad (7)$$

$$U_{X1+X3}(t) = U_{X1}(t) + extra\_in\_System\_B'(t) \quad (8)$$

where *extra_in_System_A(t)* represents the amount of unfinished work from X_3 packets in a queue whose server takes vacations during busy periods caused by the X_1 and X_2 streams. Likewise, *extra_in_System_B'(t)* represents the amount of unfinished work from X_3 packets when vacations are only during X_1 busy periods. Since busy periods caused by the X_1 stream are subintervals of busy periods caused by the combined X_1+X_2 stream, the X_3 packets in System A experience longer server vacations, and we have:

$$extra\_in\_System\_A(t) \geq extra\_in\_System\_B'(t). \quad (9)$$

Using (7)-(9) verifies (6) and concludes the proof. ❑

Intuitively interpreted, the monotonicity Lemma 3 means that the amount of blocking incurred by the (X_1+X_2) process intermixing with the X_3 process is larger than the amount incurred by the X_1 process alone mixing with the X_3 process.

These three lemmas alone are sufficient to develop some very general convexity results for unfinished work in */*/1 queues. It seems reasonable to suspect that the same three lemmas can be re-formulated in terms of *packet occupancy* (rather than unfinished work) when all packets have FIFO service. More precisely, suppose that $N_X(t)$ represents the number

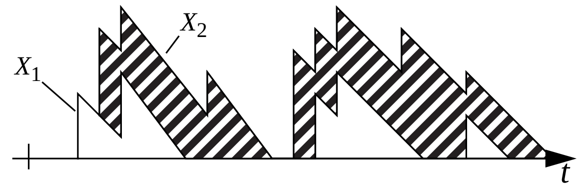

Figure 4: An example sample path of the unfinished work function $U_{X1+X2}(t)$ in a system where X_1 packets have preemptive priority.

of packets in a FIFO queueing system with input stream $X(t)$. We can define the *Occupancy Blocking Function* $\alpha_{X1,X2}(t)$ in a manner similar to (2):

$$\alpha_{X1,X2}(t) = N_{X1+X2}(t) - N_{X1}(t) - N_{X2}(t) \quad (10)$$

With this new definition of blocking in terms of packet occupancy, it can be shown that the non-negativity and symmetry properties still hold ($\alpha_{X1,X2}(t) \geq 0$, $\alpha_{X1,X2}(t) = \alpha_{X2,X1}(t)$). However, below we furnish a counterexample that demonstrates that, even under FIFO service, the *occupancy* monotonicity property does not hold for general variable length service time systems.

(Counter) Example: Under FIFO service with variable length packets, the monotonicity property of the packet occupancy blocking function does not hold, i.e., it is *not* true that $\alpha_{X1+X2,X3}(t) \geq \alpha_{X1,X3}(t)$ for all time t. The counterexample is to consider streams X_1, X_2, X_3 consisting only of one packet each, where:

-The X_3 packet enters at time 0 with service time 11.
-The X_2 packet enters at time 1 with service time 10.
-The X_1 packet enters at time 2 with service time 1.

We look at time t=4. At this time, we have: $N_{X1}(4) = 0$. When X_1 and X_2 are combined, the X_2 packet blocks the X_1 packet from being served, hence $N_{X1+X2}(4) = 2$. Likewise, $N_{X1+X3}(4)=2$, since the X_2 and X_3 packets are both long in comparison to the X_1 packet. However, because of this, when the X_3 packet is applied to a queue with the X_1 and X_2 packets, it will not generate any extra packets due to blocking. Hence, $N_{X1+X2+X3}(4) = 3$, and:

$$N_{X1+X2+X3}(4) + N_{X1}(4) = 3$$

$$< 4 = N_{X1,X2}(4) + N_{X1,X3}(4). \quad (11)$$

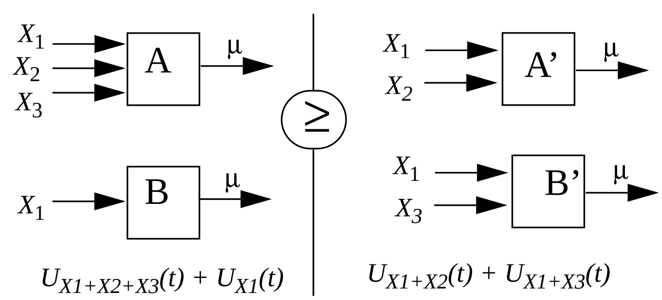

Figure 5: A queueing illustration of the monotonicity property (5) and (6).

Thus, $\alpha_{X1+X2,X3}(4) < \alpha_{X1,X3}(4)$, completing the example. ❑

Such an example relies heavily on the fact that we have variable length packets. Indeed, it can be shown that if all packets have fixed lengths L and service is non-preemptive work conserving, then the packet occupancy blocking function $\alpha_{X1,X2}(t)$ satisfies the non-negativity, symmetry, and monotonicity properties for all time.

III. Indistinguishable Inputs and Convexity

In this section and the next, we use the non-negativity, symmetry, and monotonicity properties to show that any moment of unfinished work in a */*/1 queue is a convex function of the input rate λ. To do this, we must first specify how an arbitrary input process can be parameterized by a single rate value. Here, we consider the input rate λ as a discrete quantity which is varied by adding or removing streams of the same "type" from the overall input process. We begin by developing the notion of *indistinguishable random variables*.[1]

Definition: A collection of M random variables are *indistinguishable* if:

$$p_{X1,\,X2,\,\ldots,\,XM}(x_1,\,\ldots,\,x_M) \;=\; p_{\tilde{X}1,\,\tilde{X}2,\,\ldots,\,\tilde{X}M}(x_1,\,\ldots,\,x_M) \quad (12)$$

for every $(\tilde{X}_1,\,\ldots,\,\tilde{X}_M)$ permutation of $(X_1,\ldots,X_M)$.

Thus, indistinguishable random variables exhibit a simple form of symmetry in their joint distribution functions. Definitions for random variables to be *conditionally indistinguishable* given some event ω can be similarly defined: The distributions in (12) are simply replaced by conditional distributions. It is clear that any set of independent and identically distributed (*iid*) random variables are indistinguishable. Thus, indistinguishable variables form a *wider class* than *iid* variables, and hence statements which apply to indistinguishable variables are more general. Unlike *iid* variables, however, it can be seen that if random variables $(X_1,\ldots,X_M)$ are conditionally indistinguishable given some other random variable θ, then they are indistinguishable.

We can extend this notion of indistinguishability to include random *processes* that represent packet arrival streams. The reformulation of the definitions is clear: *A collection of random processes $(X_1(t),\ldots,X_M(t))$ are indistinguishable if their joint statistics are invariant under every permutation*. Indistinguishable processes have the same properties (mentioned above) as their random variable counterparts. Below we provide three examples of indistinguishable input processes that can act as input streams to a queueing system:

Example 1: Any general */* process $X_i(t)$ independent and identically distributed over M input lines (Fig. 6a).

Example 2: Any general */* process $X(t)$ which is split into M streams by routing each packet to stream i with equal probability ($i \in \{1,\,\ldots,\,M\}$) (Fig. 6b).

[1] Our definition of "indistinguishable random variables" is identical to the established notion of "exchangeable random variables," see [8] for an interesting treatment.

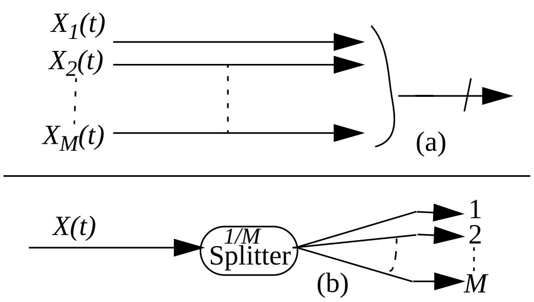

Figure 6: M indistinguishable inputs in the case of (a) the collection of *iid* */* processes $\{X_i\}$ and (b) probabilistic splitting of */* process X into M substreams.

Example 3: Any arbitrary collection of M processes $(X_1(t),\ldots,X_M(t))$ which are then randomly permuted (with each permutation equally likely).

Notice that Example 1 demonstrates the fact that *iid* inputs are indistinguishable. However, Example 2 illustrates that indistinguishable inputs form a more general class of processes by providing an important set of input streams which are not independent yet are still indistinguishable. Notice that this probabilistic routing can be modified to include "state-dependent" routing where the probability of routing to queue i depends on where the last packet was placed. The third example shows that an indistinguishable input assumption is a good a-priori model to use when an engineer is given simply a "collection of wires" from various sources, and has no a-priori way of distinguishing the process running over "wire 1" from the process running over "wire 2."

We now examine how the unfinished work in a queue changes when a sequence of indistinguishable inputs are added. Let $\theta(t)$ be an arbitrary background input process, and let $X_1(t)$ and $X_2(t)$ be two processes which are conditionally indistinguishable given $\theta(t)$. Let U_X represent the unfinished work in a queue at a particular time t^* with an input process $X(t)$ running through it (we suppress the t^* subscript in U_X for simplicity). Furthermore, let $f(u)$ represent any convex, non-decreasing function of u for $u \geq 0$. We assume that the expected value of $f(U_X)$ is well defined. (Note that expectations over functions of the form $f(u) = u^k$ represent k^{th} moments of unfinished work). The following theorem shows that incremental values of queue cost are non-decreasing with each additional input.

Theorem 1: For any particular time t^*, we have:

$$E[f(U_{\theta+X1+X2})] - E[f(U_{\theta+X1})] \geq E[f(U_{\theta+X1})] - E[f(U_\theta)] \quad (13)$$

Proof: Define the following processes:

$$\Delta_1 \;=\; U_{\theta+X1} - U_\theta \quad (14)$$

$$\Delta_2 \;=\; U_{\theta+X1+X2} - U_{\theta+X1} \quad (15)$$

We then find, by using the blocking function properties developed in the previous section:

$$\Delta_2 = U_{X2} + \beta_{\theta + X1, X2} \geq U_{X2} + \beta_{\theta, X2}$$
$$= U_{\theta + X2} - U_\theta = \tilde{\Delta}_1 \qquad (16)$$

where we have defined $\tilde{\Delta}_1 = U_{\theta + X2} - U_\theta$. Because $X_2(t)$ and $X_1(t)$ are indistinguishable given $\theta(t)$, $\tilde{\Delta}_1$ has the same distribution as Δ_1. Indeed, $\Delta_1(t)$ and $\tilde{\Delta}_1(t)$ are indistinguishable processes given $\theta(t)$. Thus, inequality (16) states that Δ_2 is a random process which is always greater than or equal to another random process which has the same distribution as Δ_1.

We now use an increasing increments property of non-decreasing, convex functions $f(u)$: For non-negative real numbers a, b, x, where $a \geq b$:

$$f(a + x) - f(a) \geq f(b + x) - f(b). \qquad (17)$$

Hence:

$$f(U_{\theta + X1} + \Delta_2) - f(U_{\theta + X1}) \geq f(U_\theta + \Delta_2) - f(U_\theta) \qquad (18)$$
$$\geq f(U_\theta + \tilde{\Delta}_1) - f(U_\theta). \qquad (19)$$

Inequality (18) follows from (17) and the fact that $U_{\theta + X1} \geq U_\theta$. Inequality (19) follows from (16). Taking expectations of the inequality above, we find:

$$E[f(U_{\theta + X1} + \Delta_2)] - E[f(U_{\theta + X1})] \geq$$
$$E[f(U_\theta + \tilde{\Delta}_1)] - E[f(U_\theta)]. \qquad (20)$$

Using the fact that $\tilde{\Delta}_1$ and Δ_1 are indistinguishable given $\theta(t)$, we can replace the $E[f(U_\theta + \tilde{\Delta}_1)]$ term on the right hand side of (20) with $E[f(U_\theta + \Delta_1)]$, which yields the desired result. ❑

The theorem above immediately suggests a convexity property of unfinished work in a work conserving queue with a collection of indistinguishable inputs. Assume we have such a collection of M streams $(X_1,...,X_M)$ which are indistinguishable given another background stream $\theta(t)$. Assume that each of the streams X_i has rate λ_δ. The total input process to the queue can then be viewed as a function of a discrete set of rates $\lambda = n\lambda_\delta$ for $n \in \{0, 1, \ldots, M\}$. Let $Ef[U(n\lambda_\delta)]$ represent the expectation of a function $f()$ of the unfinished work (at some particular time t^*) when the input process consists of stream $\theta(t)$ along with a selection of n of the M indistinguishable streams. Hence:

$$Ef[U(n\lambda_\delta)] = Ef[U_{\theta + X1 + \ldots + Xn}] \quad (0 \leq n \leq M). \qquad (21)$$

<u>Corollary 1</u>*:* At any specific time t^*, the function $Ef[U(\lambda)]$ is monotonically increasing and convex in the discrete set of rates λ ($\lambda = n\lambda_\delta$, $n \in \{0, 1, \ldots, M\}$). In particular, any moment of unfinished work is convex.

<u>Proof of Corollary 1</u>*:* Convexity of a function on a discrete set of equidistant points is equivalent to proving successive increments are monotonically increasing (Fig. 7). Hence, the statement is equivalent to:

$$Ef[U((n+2)\lambda_\delta)] - Ef[U((n+1)\lambda_\delta)]$$
$$\geq Ef[U((n+1)\lambda_\delta)] - Ef[U(n\lambda_\delta)]. \qquad (22)$$

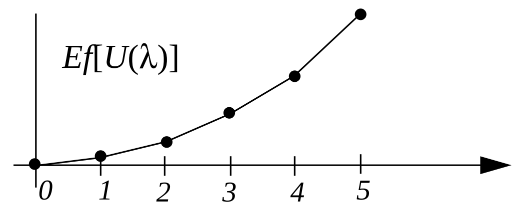

Figure 7: Convexity of unfinished work as a function of the discrete rate parameter λ.

Inequality (22) follows immediately from Theorem 1. ❑

<u>A. Waiting Times</u>: Notice that in Theorem 1 and its Corollary, expectations were taken at any particular time t^*. If inputs are stationary and yield steady state expected moments of unfinished work, we let t^* be large so that the queue is in steady state at that time. This implies that any moment of time averaged (steady state) unfinished work is a convex function of the input rate. Moreover, we can allow t^* to be a time of special interest, such as the time when a packet from the X_1 stream enters the system. In FIFO queues, the unfinished work in the system at this special time represents the amount of waiting time W that the entering packet spends in the queue before receiving service. Hence, waiting time increments are convex after the first stream is added, and we have:

<u>Corollary 2</u>*:* In FIFO queueing systems, if W represents the waiting time of packets from the indistinguishable streams, then $Ef[W(\lambda)]$ is a convex function of the discrete set of input rates $\lambda>0$ (i.e., $\lambda = n\lambda_\delta$, $n \in \{1, 2, \ldots, M\}$).

<u>B. Packet Occupancy N(t)</u>*:* Notice that the non-negativity, symmetry, and monotonicity properties of the blocking function $\beta_{X1,X2}(t)$ were the only queueing features needed to establish convexity of unfinished work $U(t)$. Now suppose that all packets have fixed lengths L, and let $N(t)$ represent the number of packets in the queueing system at time t for some arbitrary arrival process. If service in the queue is work conserving and non-preemptive, it can be shown that the occupancy blocking function $\alpha_{X1,X2}(t)$ satisfies the non-negativity, symmetry, and monotonicity properties. We can thus reformulate Theorem 1 and Corollary 1 in terms of packet occupancy. Suppose again that input streams $(X_1,..., X_M)$ are indistinguishable given background stream θ. We find:

<u>Corollary 3</u>*:* If all packets have fixed lengths L and service is non-preemptive work conserving, then at any particular time t^*, the expectation $Ef[N(\lambda)]$ is a convex function of the discrete rate λ ($\lambda \in \{0, \lambda_\delta, 2\lambda_\delta, \ldots, M\lambda_\delta\}$).

IV. Convexity over a Continuous Rate Parameter

In the previous section we dealt with streams of inputs and demonstrated convexity of unfinished work and waiting time moments as streams are removed or added. Here, we extend the theory to include input processes which are parameterized by a continuous rate variable λ. The example to keep in mind in this section is packet-by-packet *probabilistic splitting*, where individual packets from an arbitrary packet stream are sent to the queue with some probability $p \in [0, 1]$. However, the results apply to any general "infinitely divisible" input:

Definition: A packet input process $X(t)$ together with a splitting method is said to be *infinitely divisible* if:

(1) It can be split into an arbitrarily large number of substreams.

(2) Any two disjoint substreams which have the same rate are indistinguishable given the rest of the process.

(3) If x_2 is a substream which has a larger rate than another, disjoint substream x_1, then x_2 can be split into two components, one of which has the same rate as x_1.

Notice that any */* process $X(t)$ is infinitely divisible when using the probabilistic splitting method of independently including packets in a new substream i with some probability p_i. With the above definition, it can be seen that an infinitely divisible input process $X(t)$ can be written as the sum of a large number of indistinguishable substreams. Specifically, it has the property that for any $\varepsilon>0$, there exists a large integer M such that:

$$X(t) = \sum_{i=1}^{M} x_i(t) + \tilde{x}(t) \qquad (23)$$

where $(x_1(t),..., x_M(t))$ are indistinguishable substreams, each with rate λ_δ, $\tilde{x}(t)$ has rate $\tilde{\lambda}_\delta$, and $\tilde{\lambda}_\delta < \lambda_\delta < \varepsilon$.

Our definition of infinitely divisible processes above is similar in spirit to the infinitely divisible laws detailed in [9]. There, random variables decomposable into *iid* components are considered. Here, identical rate components of our process are indistinguishable but not necessarily *iid*.

We now use the blocking function to establish continuity of expected moments of unfinished work as a function of the continuous rate parameter λ. As before, these results also apply to waiting times in FIFO systems.

Again we assume that $f(u)$ is a non-decreasing convex function over $u \geq 0$. Suppose $X(t)$ is an infinitely divisible input process with total rate λ_{tot}. Suppose also that all indistinguishable component processes of $X(t)$ are conditionally indistinguishable given the background input process $\theta(t)$. Let $Ef[U(\lambda_{tot})]$ represent the expectation of a function of unfinished work at a particular time t^* in a queue with this input and background process. We assume here that $Ef[U(\lambda_{tot})]$ is finite.

Theorem 2: $Ef[U(\lambda)]$ can be written as a pure function of the continuous rate parameter λ, where $\lambda \in [0, \lambda_{tot}]$ is a rate achieved by some substream of the infinitely divisible $X(t)$ input. Furthermore, $Ef[U(\lambda)]$ is a monotonically increasing and continuous function of λ.

Proof: See Appendix. ❑

The continuity property of Theorem 2 allows us to easily establish the convexity of any moment of unfinished work (and packet waiting time) in a */*/1 queue as a function of the continuous input rate λ. Let $X(t)$ be an infinitely divisible input process, and suppose that every collection of indistinguishable components of $X(t)$ are conditionally indistinguishable given the background process $\theta(t)$. Then:

Theorem 3: At any particular time t^*, the function $Ef[U(\lambda)]$ is convex over the continuous variable $\lambda \in [0, \lambda_{tot}]$. Likewise, if service is FIFO, then $Ef[W(\lambda)]$ is also convex.

Proof: We wish to show that the function $Ef[U(\lambda)]$ always lies below its chords. Thus, for any three rates $\lambda_1 < \lambda_2 < \lambda_3$, we must verify that:

$$Ef[U(\lambda_2)] \leq Ef[U(\lambda_1)] + (\lambda_2 - \lambda_1)\frac{(Ef[U(\lambda_3)] - Ef[U(\lambda_1)])}{(\lambda_3 - \lambda_1)} \qquad (24)$$

We know from Theorem 1 and Corollary 1 in Section III that the unfinished work function is convex over a discrete set of rates when the input process is characterized by a finite set of M indistinguishable streams $(x_1,..., x_M)$. We therefore consider a discretization of the rate axis by considering the subprocesses $(x_1,..., x_M)$ of the infinitely divisible process $X(t)$, where each x_i has a small rate δ. In this discretization, we have rates:

$$\tilde{\lambda}_1 = k_1\delta \ , \quad \tilde{\lambda}_2 = k_2\delta \ , \quad \tilde{\lambda}_3 = k_3\delta \qquad (25)$$

where the rates $(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3)$ can be made arbitrarily close to their counterparts $(\lambda_1, \lambda_2, \lambda_3)$ by choosing an appropriately small value of δ. Now, from the discrete convexity result, we know:

$$Ef[U(\tilde{\lambda}_2)] \leq Ef[U(\tilde{\lambda}_1)] + (\tilde{\lambda}_2 - \tilde{\lambda}_1)\frac{(Ef[U(\tilde{\lambda}_3)] - Ef[U(\tilde{\lambda}_1)])}{(\tilde{\lambda}_3 - \tilde{\lambda}_1)} \qquad (26)$$

By continuity of the $Ef[U(\lambda)]$ function, we can choose the discretization unit δ to be small enough so that the right hand side of (26) is arbitrarily close to the right hand side of the (currently unproven) inequality (24). Simultaneously, we can ensure that the left hand sides of the two inequalities are arbitrarily close. Thus, the known inequality (26) for the discretized inputs implies inequality (24) for the infinitely divisible input. We thus have convexity of unfinished work at any point in time, which also implies convexity of waiting time in FIFO systems. ❑

V. Multiple Queues in Parallel

We now consider the system of N queues in parallel as shown in Fig. 8. The servers of each queue have linespeeds μ_i and arbitrary background packet input processes $\theta_i(t)$. An arbitrary input process $X(t)$ also enters the system, and $X(t)$ is rate-controllable in that a router can split $X(t)$ into substreams of smaller rate. These substreams can be distributed according to an N-tuple rate vector $(\lambda_1,..., \lambda_N)$ over the multiple queues.

We consider both the case when $X(t)$ is an infinitely divisible process (as in packet-based probabilistic splitting), and the case when $X(t)$ is composed of a finite collection of M indistinguishable streams. The problem in both cases is to route the substreams by forming an optimal rate vector that minimizes some network cost function. We assume the cost function is a weighted summation of unfinished work and/or waiting time moments in the queues. Specifically, we let $\{f_i(u)\}$ be a collection of convex, non-decreasing functions on $u \geq 0$. Suppose that the queues reach some steady state behavior, and let $U_i(\lambda_i)$ represent the steady state value of unfinished work in queue i when an input stream of rate λ_i is applied. Let $W_i(\lambda_i)$ represent the steady state waiting time for queue i.

Theorem 4: If queues are work conserving and $X(t)$ is either a finitely or infinitely rate divisible process given $\{\theta_i(t)\}$, then:

(a) Cost functions of the form

$$\Phi(\lambda_1, \ldots, \lambda_N) = \sum_{k=1}^{N} E f_k[U_k(\lambda_k)] \tag{27}$$

are convex in the multivariable rate vector $(\lambda_1,..., \lambda_N)$.

(b) If service is FIFO, then cost functions of the form

$$\Phi(\lambda_1, \ldots, \lambda_N) = \sum_{k=1}^{N} \lambda_k E f_k[W_k(\lambda_k)] \tag{28}$$

are convex.

(c) If service is FIFO and $N_k(\lambda_k)$ represents the number of packets in queue k in steady state, then cost functions of the following form are convex:

$$\Phi(\lambda_1, \ldots, \lambda_N) = \sum_{k=1}^{N} a_k E[N_k(\lambda_k)] \quad (\{a_k\} \geq 0). \tag{29}$$

Proof: Since $Ef_k[W_k(\lambda_k)]$ is convex and non-decreasing for $\lambda_k>0$, the function $\lambda_k Ef_k[W_k(\lambda_k)]$ is convex on $\lambda_k \geq 0$. Thus, the cost functions in (a) and (b) are summations of convex functions, so they are convex. Part (c) follows from (b) and noting that, from Little's Theorem, $E[N] = \lambda E[W]$. ❑

Convexity of the cost function $\Phi(\lambda_1, \ldots, \lambda_N)$ can be used to develop optimal rate distributions $(\lambda_1^o, \ldots, \lambda_N^o)$ over the simplex constraint $\lambda_1+...+\lambda_N=\lambda_{tot}$. For symmetric cost functions, which arise when the background processes $\{\theta_i(t)\}$ and the linespeeds $\{\mu_i\}$ are the same for all queues $i \in \{1, \ldots, N\}$, the optimal solution is particularly simple. It is clear in this case that the uniform distribution $(\lambda_1^o, \ldots, \lambda_N^o) = (\lambda_{tot}/N, \ldots, \lambda_{tot}/N)$ (or as near to this as can be achieved) is optimal and minimizes cost. Thus, in the symmetric case we want to spread the total input stream evenly amongst all of the queues in order to take full advantage of the bandwidth that each queue provides. In the asymmetric case when background streams and linespeed processes are not the same, the optimal rate vector deviates from the uniform allocation to favor queues with faster linespeeds and/or less background traffic.

Let us suppose the cost function $\Phi(\lambda_1, \ldots, \lambda_N)$ is known. Convexity of Φ tells us that any local minimum of the cost function we find must also be a global minimum. It also tells us a great deal more, as we illustrate for both finitely divisible

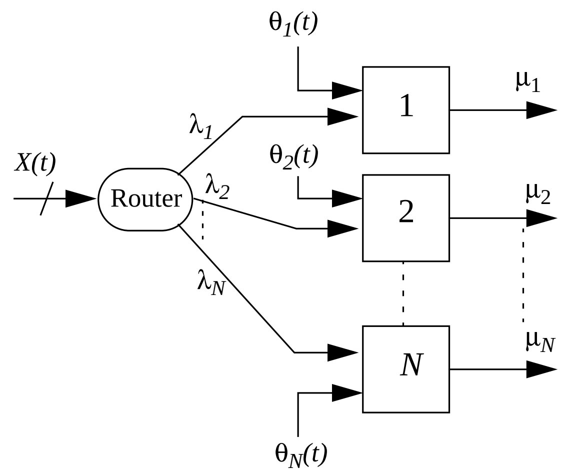

Figure 8: Multiple queues in parallel with different background processes $\{\theta_i(t)\}$ and server rates $\{\mu_i\}$.

and infinitely divisible inputs $X(t)$ below:

A. (Stream-Based) Finitely Divisible X(t): Here we assume that the input $X(t)$ is a finite collection of M streams, which are indistinguishable given the background processes $\{\theta_i(t)\}$. We want to distribute the streams over the N queues available. We can thus write the cost function Φ as a function of an integer N-tuple $(k_1, \ldots, k_N)$ (rather than a rate N-tuple) which describes the number of streams we route to each queue. If the queues are weighted the same and have indistinguishable background inputs and identical linespeeds, then the cost function $\Phi(k_1, \ldots, k_N)$ is convex symmetric and the optimal solution is to allocate $\lceil M/N \rceil$ streams to $(M)mod(N)$ of the queues, and $\lfloor M/N \rfloor$ streams to the remaining queues. In the non-symmetric case, we must consider other allocations and test them by evaluating the cost function.

Theorem 5: Given a convex cost function $\Phi(k_1, \ldots, k_N)$ of the form specified in Theorem 4, the optimal allocation vector can be obtained by sequentially adding streams, greedily choosing at each iteration the queue which increases the total cost Φ the least. This yields a cost-minimizing vector $(k_1^o, \ldots, k_N^o)$ after M+N-1 evaluations/estimations of the cost function.

Proof: The theorem is clearly true for M=1 stream. We assume that it is true for M=k streams, and by induction prove it holds for M=k+1.

Let $(k_1^o, \ldots, k_N^o)$ represent the optimal allocation vector for M=k streams, which is obtained by the sequentially greedy algorithm. We thus have $\sum_i k_i^o = k$.

Now we add an additional stream in a greedy manner by placing it in the queue which increases the cost function the least. Without loss of generality, we assume this queue is queue 1, and we have a new allocation vector $(k_1^o + 1, k_2^o, \ldots, k_N^o)$. Suppose there is some other vector $(\tilde{k}_1, \ldots, \tilde{k}_N)$ whose elements sum to k+1, such that $\Phi(k_1^o + 1, k_2^o, \ldots, k_N^o) > \Phi(\tilde{k}_1, \ldots, \tilde{k}_N)$.

Case 1: $\tilde{k}_1 \geq k_1^o + 1$. In this case, we take away an input stream from the first entry of both vectors. This effects only the queue 1 term $Ef_1[k]$ in the cost function. Because this function is convex and non-decreasing, $Ef_1[k_1^o + 1]$ decreases by less than or equal to the amount that $Ef_1[\tilde{k}_1]$ decreases. Hence, it must be that:
$\Phi(k_1^o, \ldots, k_N^o) > \Phi(\tilde{k}_1 - 1, \ldots, \tilde{k}_N)$, which contradicts the fact that the $(k_1^o, \ldots, k_N^o)$ is optimal for M=k streams. ❑

Case 2: $\tilde{k}_1 < k_1^o + 1$. In this case, there exists some queue j such that $\tilde{k}_j > k_j^o$. We take the additional input we added to queue 1 and move it to queue j, forming a new vector $(k_1^o, \ldots, k_j^o + 1, \ldots, k_N^o)$. Notice that this change cannot decrease the cost function, since this input was originally added greedily to queue 1. We now have $\tilde{k}_j \geq k_j^o + 1$, which reduces the problem to Case 1. ❑

Thus, the sequentially greedy algorithm is optimal. It can be implemented with M+N-1 evaluations of the cost function by keeping a record of the N-1 queue increment values for the N-1 queues not chosen at each step. ❑

Example: We consider the system of Fig. 8 when there are N=4 queues and the input $X(t)$ consists of 200 independent streams that produce packets periodically every P seconds. Packets have a fixed length of L bits. The streams are unsynchronized, and hence 200 packets arrive in a uniformly distributed fashion over any time interval of length P. Such input streams are models for continuous bit rate traffic, such as digital voice or video data.

The problem is to distribute the streams over the 4 queues while minimizing the cost function, which we take to be the total expected number of packets in the system (this is the cost function of Theorem 4c). We first assume all server speeds μ_i are the same, and all background streams $\theta_i(t)$ are indistinguishable. In this case, we immediately know the optimal stream allocation vector is (50, 50, 50, 50).

Now suppose that we have server speeds (2, 1, 1, 1) and that there are 10 background streams of the same type at queue 4. In this asymmetric case, we must use the cost function to determine optimal allocation using the sequentially greedy method. The complementary occupancy distribution in a single queue with K inputs of period P, length L, and server rate μ has been derived explicitly in [5]. The result is:

$$Q_n\left(\frac{L}{\mu P}, K\right) = \sum_{i=1}^{K-n} \binom{K}{i+n} \left(\frac{iL}{\mu P}\right)^{i+n} \left(1 - \frac{iL}{\mu P}\right) \left(\frac{\mu P/L - K + n}{\mu P/L - i}\right)$$
$$(0 \leq n \leq K-1). \qquad (30)$$

The expected number of packets is hence:

$$\bar{N}\left(\frac{L}{\mu P}, K\right) = \sum_{n=0}^{K-1} Q_n\left(\frac{L}{\mu P}, K\right). \qquad (31)$$

We therefore use the greedy algorithm with cost function:

$$\Phi(K_1, K_2, K_3, K_4) = \sum_{i=1}^{4} \bar{N}\left(\frac{L}{\mu_i P}, K_i\right). \qquad (32)$$

Using L=1, P=75, we find that the optimal allocation vector is: (100, 37, 37, 26). From this solution, we see that--as expected because of the 10 background streams in queue 4--there are approximately 10 more streams allocated to queues 2 and 3 than queue 4. Interestingly, the speed of queue 1 is twice that of the other queues, although, due to statistical multiplexing gains, the number of streams it is allocated is more than twice the number allocated to the others.

B. (Packet Based) Infinitely Divisible $X(t)$: Here we consider an infinitely divisible process $X(t)$ with total rate λ_{tot} as an input to the system of Fig. 8. The problem is to optimally distribute the total rate over the N queues so as to minimize a cost function $\Phi(\lambda_1, \ldots, \lambda_N)$. We assume the cost function is of one of the forms specified in Theorem 4. Each of these had the structure:

$$\Phi(\lambda_1, \ldots, \lambda_N) = g_1(\lambda_1) + \ldots + g_N(\lambda_N) \qquad (33)$$

for some convex, non-decreasing functions $g_i(\lambda_i)$.

If background inputs are indistinguishable, and if all queues are weighted the same in the cost function, then Φ is convex symmetric and the optimal rate allocation is (λ_{tot}/N,..., λ_{tot}/N). Otherwise, we take advantage of the structure of the Φ function to determine the optimal solution.

Each of the functions $g_i(\lambda_i)$ is non-decreasing and convex on some interval $(0, \tilde{\lambda}_i)$. It can be shown that these properties ensure $g_i(\lambda_i)$ is right-differentiable. They are also sufficient to establish the correctness of a Lagrange Multipliers approach to cost minimization. Given the Lagrangian:

$$L(\lambda_1, \ldots, \lambda_N, \gamma) = \Phi(\lambda_1, \ldots, \lambda_N) + \gamma\left(\lambda_{tot} - \sum_{i=1}^{N} \lambda_i\right) \qquad (34)$$

where γ is the Lagrange Multiplier, we differentiate (from the right) with respect to λ_i to obtain

$$\frac{d}{d\lambda_i} g_i(\lambda_i) = \gamma \qquad \text{for all } i \in \{1, \ldots, N\} \qquad (35)$$

subject to the simplex constraint λ_1+...+λ_N = λ_{tot}. Fig. 9 illustrates this solution. If we define:

$$\lambda_i(\gamma) = \text{Largest } \lambda \text{ such that } g_i'(\lambda) \leq \gamma \qquad (36)$$

then from the figure we see that we increase the value of γ until $\lambda_1(\gamma) + \ldots + \lambda_N(\gamma) = \lambda_{tot}$. The resulting rate vector yields the optimal solution.

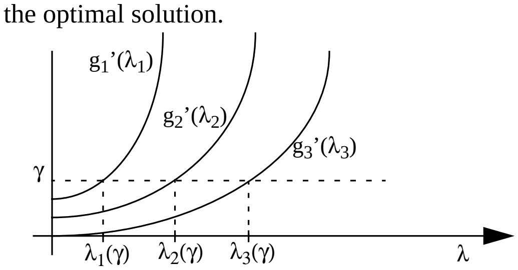

Figure 9: A sample set of $\frac{d}{d\lambda_i} g_i(\lambda_i)$ curves.

Notice that a fast bisection type algorithm can be developed to find this optimal rate vector. First, two bracketing values γ_{Low} and γ_{Hi} are found which yield $\lambda_1(\gamma) + \ldots + \lambda_N(\gamma)$ values above and below λ_{tot}, respectively. The bisection routine proceeds as usual until the rate vector converges to a solution within acceptable error limits.

Example: The following simple example for systems with memoryless arrivals and packet lengths demonstrates the method. Suppose, for simplicity, that there are no background arrivals $\{\theta_i(t)\}$. Let the arrival process be Poisson with rate λ, and the packet length process be *iid* and memoryless with an average packet length of 1 unit. We assume that we are using probabilistic Bernoulli routing where each packet is routed independently of the next. Let $\overline{N}_i$ represent the average number of packets in queue *i*, and define cost function:

$$\Phi(\lambda_1, \ldots, \lambda_N) = \sum_i a_i \overline{N}_i(\lambda_i) \tag{37}$$

where

$$g_i(\lambda_i) = a_i \overline{N}_i(\lambda_i). \tag{38}$$

The expected number of packets in an M/M/1 queue is:

$$\overline{N}_i(\lambda_i) = \frac{\lambda_i/\mu_i}{1 - \lambda_i/\mu_i} \ . \tag{39}$$

Multiplying the above equation by a_i, taking derivatives with respect to λ_i and setting the result equal to γ for all *i*, as well as considering the constraint $\lambda_1 + \ldots + \lambda_N = \lambda_{\text{tot}}$, we find for all $k \in \{1, \ldots, N\}$:

$$\lambda_k = \mu_k + \frac{\sqrt{a_k \mu_k}\left(\lambda_{tot} - \sum_i \mu_i\right)}{\sum_i \sqrt{a_i \mu_i}} \ . \tag{40}$$

The above equation is a bit deceptive, in that the summations are taken over all *i* for which λ_i is positive. The positive λ_i's are determined by first assuming that all are positive and applying the above equation. If any of the λ_i's found are zero or negative, these λ_i's are set to zero and the calculation is repeated using the remaining subset of λ_i's.

This approach to optimal rate allocation is similar to the convex optimization routines described in [6]. There, the authors address packet routing in general mesh networks when input streams can be continuously split according to a rate parameter. They pre-suppose some convex cost function at each node of the network, which (as an idealization) is completely a function of the overall rate routed to that node. Here, we have considered cost functions which reflect the actual queuing congestion at each node when a general */* input is applied. We have established that, for a simple network consisting of *N* parallel queues, the cost functions at each queue are continuous and non-decreasing in the rate parameter, and they are indeed convex.

VI. Time-Varying Server Speeds

Here we consider the system of Fig. 1 when the constant server of rate μ is replaced by a time varying server of rate $\mu(t)$. Define $U^{(bits)}(t)$ to be the amount of unprocessed data (in units of bits) in the system at time *t*. This value is different from the unfinished work *U* previously considered--which had units of *time* ($U^{(time)}(t)$). Note that $U^{(bits)}(t) = \mu U^{(time)}(t)$ for constant server rate systems, while $U^{(time)}(t)$ is not causally known from the system state for time varying systems.

Characteristics of a $U^{(bits)}(t)$ sample path are similar to those illustrated in Fig. 2 for constant server systems, with the exception that the $U^{(bits)}(t)$ function decreases with a time varying slope $-\mu(t)$. We can define the unfinished data blocking function:

$$\beta^{(bits)}{}_{X1,X2}(t) = U^{(bits)}{}_{X1+X2}(t) - U^{(bits)}{}_{X1}(t) - U^{(bits)}{}_{X2}(t). \tag{41}$$

By repeating the arguments of section II in terms of this new blocking function, we can establish that the non-negativity, symmetry, and monotonicity properties hold for $\beta^{(bits)}{}_{X1,X2}(t)$. Likewise, if all packets have fixed lengths *L* and service is non-preemptive, it can be shown that the occupancy blocking function $\alpha_{X1,X2}(t)$ satisfies these three properties in this time-varying server setting.

Consequently, given a collection of *N* queues with background input processes $\{\theta_i(t)\}$ and server rate processes $\{\mu_i(t)\}$, together with a (finitely or infinitely distributable) input $X(t)$, we can establish:

Theorem 6: If the indistinguishable components of $X(t)$ are conditionally indistinguishable given $\{\theta_i(t)\}$ and $\{\mu_i(t)\}$, then $\sum E f_i[U_i^{(bits)}(\lambda_i)]$ is convex in the rate vector $(\lambda_1,\ldots,\lambda_N)$. If all packets have a fixed length of *L* bits and service is non-preemptive, then $\sum E f_i[N_i(\lambda_i)]$ is convex in the rate vector. ❑

Recall from Little's Theorem that if the expected waiting time $E[W(\lambda)]$ is convex in λ, then so is the expected packet occupancy $E[N(\lambda)]$. However, the converse implication does not follow. Indeed, below we provide a (counter) example which illustrates that--even for fixed length packets under FIFO service--waiting times are not necessarily convex for time varying servers.

(Counter) Example: Consider identical input processes X_1, X_2, X_3 which produce a single packet of length L=1 periodically at times {0, 3, 6, 9,...}. Let the server rate be periodic of period 3 with $\mu(t)=1$ for $t \in [0, 2]$ and $\mu(t)=100$ for $t \in (2, 3)$. Then $E[W_{X1}]=1$, $E[W_{X1+X2}]=1.5$, and $E[W_{X1+X2+X3}]=1.67$. Clearly the increment in average waiting time when stream X_2 is added is *larger* than the successive increment when stream X_3 is added. Hence, waiting time is not convex in this time-varying server setting. However, notice that minimizing $\overline{W}_{tot}$ in a parallel queue configuration (Fig. 8) is accomplished by minimizing $\overline{N}_{tot}$ (since $\overline{N}_{tot} = \lambda_{tot}\overline{W}_{tot}$). For fixed length packets, Theorem 6 ensures this is a convex optimization even for time varying servers.

VII. CONCLUSIONS

We have developed general convexity results for */*/1 queues using a new function of two packet stream inputs called the *blocking function*. Non-negativity, Symmetry, and Monotonicity properties of the blocking function were established. These properties proved to be valuable tools for establishing convexity of unfinished work and waiting time moments ($Ef[U(\lambda)]$ and $Ef[W(\lambda)]$) in terms of both a discrete and a continuous input rate λ.

We then addressed both stream-based and packet-based routing of general inputs over a collection of N parallel queues with arbitrary background inputs and different linespeeds. Optimal routing algorithms were developed utilizing the convexity results.

This convexity theory can be extended to address more complex variants of the parallel queue problem. One might consider a case when we have a collection of K sets of indistinguishable inputs, where each set categorizes a different type of input process. For example, set S_1 could contain multiple indistinguishable inputs of the "bursty" type, while set S_2 contains indistinguishable inputs of the "continuous bit rate" type. This variation of the problem is closely related to the NP-complete bin packing problem. It would also be interesting to explore convexity and optimal routing in more general mesh networks using these techniques. Such an approach could perhaps establish the validity of known convex optimization routines, as well as provide insights into developing new ones.

APPENDIX:

Here we show that $Ef[U(\lambda)]$ is a continuous, increasing function of λ (Theorem 2 of Section IV). We utilize the following facts about convex, non-decreasing functions:

<u>*Fact 1*</u>*:* If $f(u)$ is non-decreasing and convex, then for any fixed $a \geq 0$ there is a function $g(a, x)$ such that $f(a+x) = f(a) + g(a, x)$, where $g(a, x)$ is a convex, non-decreasing function of x for $x \geq 0$.

<u>*Fact 2*</u>*:* Any convex, non-decreasing function $g(x)$ with $g(0)=0$ has the property that $g(x_1 + x_2) \geq g(x_1) + g(x_2)$ for any $x_1, x_2 \geq 0$.

<u>*Proof of Theorem 2*</u>*:* It is straightforward to verify that $Ef[U(\lambda)]$ is a pure, monotonically increasing function of λ. Here we prove that the function is continuous from the right. Left continuity can be proven in a similar manner.

Take any λ in the set of achievable rates. We show that:

$$\lim_{\delta \to 0} Ef[U(\lambda+\delta)] = Ef[U(\lambda)] \tag{41}$$

where δ is the rate of a component process of $X(t)$ which we let get arbitrarily small. By monotonicity, if δ decreases to zero, then $Ef[U(\lambda+\delta)]-Ef[U(\lambda)]$ decreases toward some limit ε, where $\varepsilon \geq 0$. Suppose now that this inequality is strict. We reach a contradiction.

Consider disjoint component streams $(x_1,..., x_M)$, each x_i of rate δ, for some yet-to-be-determined δ and M. We assume that these M substreams are disjoint from another substream θ of rate λ, all of which are components of the entire process $X(t)$. Let $U_{\theta+x1+...+xM}$ represent the unfinished work in the system at some particular time t^*, with input processes $(\theta, x_1,...,x_M)$. From the definition of the blocking function, we have:

$$U_{\theta+x1+\ldots+xM} = U_{\theta+x1+\ldots+x(M-1)} + U_{xM} + \beta_{\theta+x1+\ldots+x(M-1),\, xM}. \tag{42}$$

By recursively iterating (42), we find:

$$U_{\theta+x1+\ldots+xM} = U_\theta + \sum_{i=1}^{M} U_{xi} + \sum_{k=0}^{M-1} \beta_{\theta+x1+\ldots+xk,\, x(k+1)}. \tag{43}$$

Applying the monotonicity property of the blocking function to (43), we obtain:

$$U_{\theta+x1+\ldots+xM} \geq U_\theta + \sum_{i=1}^{M} [U_{xi} + \beta_{\theta,\, xi}]. \tag{44}$$

Now applying the monotonically increasing, convex function $f(u)$ to both sides of (44) and writing $f(U_\theta+x) = f(U_\theta) + g(U_\theta, x)$ (from *Fact 1*), we have:

$$f(U_{\theta+x1+\ldots+xM}) \geq f(U_\theta) + g\left(U_\theta, \sum_{i=1}^{M}[U_{xi} + \beta_{\theta,\, xi}]\right) \tag{45}$$

$$\geq f(U_\theta) + \sum_{i=1}^{M} g(U_\theta, [U_{xi} + \beta_{\theta,\, xi}]). \tag{46}$$

Inequality (46) holds because $g(U, x)$ is a convex function of x and is zero at x=0 (*Fact 2*). Now notice that $Ef[U(\lambda+\delta)]-Ef[U(\lambda)]= Ef[U_{\theta+xi}]-Ef[U_\theta] =E[g(U_\theta,U_{xi}+\beta_{\theta,xi})]$ for any θ substream of rate λ, and any disjoint x_i substream of rate δ. Hence, by assumption:

$$E[g(U_\theta, U_{xi} + \beta_{\theta,\, xi})] \geq \varepsilon > 0. \tag{47}$$

Taking expectations of (46) and using (47), we find:

$$Ef[U_{\theta+x1+\ldots+xM}] \geq Ef[U_\theta] + M\varepsilon. \tag{48}$$

Inequality (48) above holds whenever $\theta+x_1+...+x_M$ is a substream of the entire, infinitely divisible process $X(t)$. We now choose M large enough so that $M\varepsilon$ is greater than the expectation of $f(U)$ when the entire input $X(t)$ is applied, i.e., $M\varepsilon>Ef[U_X]$. However, we choose a rate δ for each of the x_i substreams that is small enough so that we can still find M such substreams to ensure $\theta+x_1+...+x_M$ is still a component process of X(t). This implies that, from (48), $Ef[U_{\theta+x1+\ldots+xM}] > Ef[U_X]$, which contradicts monotonicity. Hence, ε=0, (41) holds, and the theorem is proven. ❑

REFERENCES:

[1] E. Altman, B. Gaujal, and A. Hordijk, "Multimodularity, Convexity and Optimization Properties," INRIA report, June 1997.

[2] M. J. Neely and C. E. Rohrs, "Equivalent models and analysis for multi-stage tree networks of deterministic service time queues," Allerton Conference Proceedings, 2000.

[3] M.J. Neely with C.E. Rohrs, *Queue Occupancy in Single Server, Deterministic Service Time Tree Networks*. Masters Thesis, MIT LIDS, March 1999.

[4] L. Gun, A. Jean-Marie, A.M. Makowski, and Tedijanto, "Convexity results for parallel queues with Bernoulli routing," ISR Technical Report, University of Maryland 1990.

[5] J.W. Roberts and J.T. Virtamo, "The superposition of periodic cell arrival streams in an ATM multiplexer," IEEE Trans. Comm, vol. 39, no.2. February 1991.

[6] D.P. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[7] Robert G. Gallager, *Discrete Stochastic Processes*. Boston: Kluwer Academic Publishers, 1996.

[8] Sheldon Ross, *Stochastic Processes*. New York: John Wiley & Sons, Inc., 1996.

[9] Daniel W. Stroock, *Probability Theory, an Analytic View*. Cambridge University Press, 1993.