Round-Robin Scheduling for Max-Min Fairness in Data Networks

Ellen L. Hahne

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

This paper studies a simple strategy, proposed independently by Gallager [1] and Katevenis [2], for fairly allocating link capacity in a point-to-point packet network with virtual circuit routing. Each link offers its packet transmission slots to its user sessions by polling them in round-robin order. In addition, window flow control is used to prevent excessive packet queues at the network nodes. As the window size increases, the session throughput rates are shown to approach limits that are perfectly fair in the max-min sense. That is, the smallest session rate in the network is as large as possible and, subject to that constraint, the second-smallest session rate is as large as possible, etc. If each session has periodic input (perhaps with jitter) or has such heavy demand that packets are always waiting to enter the network, then a finite window size suffices to produce perfectly fair throughput rates.

The round-robin method is considerably simpler than earlier strategies for achieving global fairness. The fair session rates are not explicitly computed, and the only overhead communication is that required for the window acknowledgments. The main drawback is that large windows are needed to achieve even approximately fair throughputs in some (hopefully rare) situations, and large windows permit large cross-network delays. Fortunately, the round-robin method offers other throughput guarantees that, while falling short of perfect fairness, do apply even for sessions with small windows. Such sessions are promised reasonable bounds on their cross-network packet delay as well.

This work was supported by the Defense Advanced Research Projects Agency (Contract ONR/N00014-84-K-0357), the Army Research Office (Contract DAAG29-84-K-0005), the National Science Foundation (Grant NSF-ECS-8310698), the AT&T Doctoral Scholarship Program, the Vinton Hayes Fellowship Program, and AT&T Bell Laboratories. This paper is based on a Ph.D. dissertation submitted to MIT's Department of Electrical Engineering and Computer Science in December 1986.

This paper appears in *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, September 1991, pp. 1024-1039.

1. INTRODUCTION

1.1 Background

Consider a data communication network consisting of store-and-forward switches (*nodes*) joined by point-to-point communication channels (*links*). Each network user (*session*) is assigned a fixed path (*virtual circuit*) through the network, and data for the session are sent in manageable parcels (*packets*) along this path. In such a network, occasional surges in user demand can overload network links, causing packet queues to build up in network nodes. These queues may eventually overflow the nodes' storage space, or the delay of acknowledgments may cause transmitters to assume that data were lost. These problems result in wasteful retransmissions that effectively reduce the capacity of the network. Flow control procedures attempt to prevent or alleviate this degradation by regulating the appropriate traffic sources. Reference [3] discusses many of the flow control techniques that have been proposed in the literature.

One such scheme is the *window method* [3]. This technique limits the number of packets for each session that have been transmitted but for which acknowledgments have not yet been received. The maximum permissible number of outstanding packets is called the *window size*. In the *end-to-end* method, a single window is applied to all of a session's traffic, and the session's destination node sends an acknowledgment to the origin node whenever a packet is claimed by the session's sink. In the *link-by-link* or *node-by-node* method, the session has a separate window for its traffic over each link, and whenever a packet is transmitted from a node, that node sends an acknowledgment to the packet's preceding node. The window method is described here because it is a component of several more elaborate strategies to be discussed later.

It would be desirable for flow control procedures to regulate network inputs so as to grant each session a fair throughput rate. Reference [3] explains that many proposed flow control methods are unfair. Several studies have addressed the issue of throughput fairness, however, and these will now be briefly discussed. Another literature survey on this topic can be found in [4].

The problem of achieving fair throughput rates can be broken into three parts. First the fairness objective must be formulated precisely. Then the fair session rates must be determined. Finally, these rates must be enforced. References [5], [6], [7: Section 5.2], [8], [9: Chapter 3], [10: Sections 4 and 6.2], and [11] - [18] have objectives of roughly the same form, viz., maximization of a sum of terms, one for each session. Each term measures the satisfaction of a session as a

function of its throughput rate and possibly its average packet delay. Many different satisfaction functions have been suggested. Another fairness approach, called *max-min flow control* or *bottleneck flow control*, is used in various forms in [7: Section 3], [9: Chapter 4], [10: Chapter 3], and [19] - [25]. Only the simplest version of this objective, viz., Hayden's [9], will be defined here. To satisfy the max-min flow criterion, the smallest session rate in the network must be as large as possible. Subject to this constraint, the second-smallest session rate must be as large as possible, etc. Given a network with its link capacities and a set of sessions with their routes and their maximum possible transmission rates, there is a unique set of session rates that satisfies the max-min conditions. The max-min flow criterion will be taken as the definition of throughput fairness in this paper. Most of the studies mentioned in this paragraph also develop algorithms for computing session rates that are fair according to the various criteria. Many of these algorithms are meant to be implemented in a distributed manner.

Beyond the issues of defining and computing fair rates is the problem of enforcing them. Several methods have been suggested in the literature. References [5], [6], [7: Section 5.2], [8: Section 4.7], [10: Chapter 5], and [12] use window flow control and adjust the sessions' window sizes to achieve the desired rates. References [8: Section 4.2], [9: Chapter 5], [10: Chapter 3], [19], [22], and [24] consider a session input control that produces packet lengths proportional to the session's assigned rate. The time between packet admissions is constant. This model is particularly meaningful for packetized voice traffic: it represents the output of a variable rate vocoder [19]. Reference [23] takes the opposite approach, called *metering*. Time is divided into control intervals whose length is inversely proportional to a session's target rate. The session is permitted to inject some fixed quantity of data into the network during one control interval. A fourth approach, taken in [15], [16], [26], and [27], is to schedule the use of the links among the various sessions. These studies assume that window flow control is also used, but it is primarily the schedule parameters rather than the window sizes that are adjusted to achieve the desired session rates.

This paper studies the following method proposed independently by Gallager [1] and Katevenis [2] for max-min fair flow control. (The technique is very similar to that in [28] - [30].) Each link offers its packet transmission slots to its users by polling them in round-robin order. If a session is offered a chance to use a link slot but has no packets ready, then that same slot is offered to the next session, and perhaps the next, etc., until a ready session is found. In each pass of a link's round robin, a session may transmit only one packet. The round-robin schedulers for the various links are uncoordinated. In order to prevent excessive queues at the network nodes, window flow control is also employed. The

principal contribution of this paper is a proof that round-robin scheduling with windows can indeed be used to achieve max-min throughput fairness.

The main advantage of the round-robin method is its simplicity. The desired rates are never explicitly computed, as they are for most other fair flow control schemes. The only overhead communication is that required for the window acknowledgments. The window sizes need not be adjusted as network conditions change. The round-robin operation is not difficult to implement [2]. An obvious price paid for this simplicity is a lack of flexibility. The strategy is coupled to the max-min criterion and probably cannot be adapted to fairness objectives substantially different from max-min. (Session priorities *can* be implemented, however, by allowing some sessions to transmit more than one packet over a link in each polling cycle [31-36].) This paper points out another drawback, viz., that large windows are needed to ensure throughput fairness in some (hopefully rare) situations. Large windows require lots of nodal storage, permit large cross-network delays, and encourage slow convergence of the session throughput rates to their new values when network conditions change. Fortunately, the round-robin method offers other throughput guarantees that, while falling short of perfect fairness, do apply even for sessions with small windows; such sessions are promised reasonable bounds on their cross-network packet delay as well. For completeness, this paper includes these throughput and delay bounds.

1.2 Overview

The system model assumed throughout this paper is described in Section 2. The network is composed of switching nodes connected by point-to-point links with equal capacities. Uniform packet lengths are also assumed. The network supports a fixed set of sessions with virtual circuit routing. Sessions are served at a link on a round-robin basis. This paper assumes link-by-link window flow control. (An end-to-end model is considered in [37].) The round-robin service discipline is modeled *exactly*, as are the link-by-link windows (which are effectively tandem finite queues). Several packet arrival models are considered, but packets are assumed to depart as soon as they reach their destination nodes. The resulting system model is considerably more realistic than a standard network-of-queues model [38-40]. Due to the complexity of our model, only a worst-case analysis is possible, and most of our performance bounds are probably not sharp.

The max-min flow criterion, which is taken as the definition of throughput fairness, is described in Section 3. The heart of this paper consists of three theorems on max-min fairness, one assuming Bernoulli packet arrivals and two for

more regular demand. Theorem 3 (Section 6) asserts that, for Bernoulli sources, the session throughput rates approach the max-min fair rates as the window size increases. If each session has periodic input (perhaps with jitter) or has such heavy demand that packets are always waiting to enter the network, then the traffic admitted to the network will be more regular. For this smoother demand model, Theorem 2 (Section 5) claims that the long-term average throughputs equal the fair rates if the window size is at least a certain value $W_{\min}$. Moreover, under the same demand and window size assumptions, Theorem 1 (Section 4) shows that the throughput of a session during any finite interval is within a constant number of packets of the max-min fair amount (regardless of the length of the interval). The constant is proportional to the window size, because the session throughputs cannot stabilize until the lengths of the packet queues stabilize. According to Theorem 2, a steady state is eventually reached; thereafter, the unfairness of a session's throughput over any interval is less than another constant number of packets. This constant is independent of the window size. References [41] and [1] contain several corollaries of Theorem 2 dealing with steady-state queue lengths. (See [42] for a similar style of traffic model.)

The minimum window size $W_{\min}$ for which Theorem 2 proves perfect fairness of the long-term average throughputs is absurdly large and grows exponentially with the number of sessions using the network. Moreover, recall that for the Bernoulli demand result, no finite window size is sufficient. These analyses are meant to show the *existence* of fair flows with the round-robin method; they are *not* meant to precisely quantify the window requirements. Hence one wonders how large the windows must actually be to achieve at least approximate fairness in practice. Section 5 includes an example where the session throughput rates are quite unfair unless large windows are used. [1] Another perspective is to *fix* a small, practical window size, then ask how unfair the resulting throughputs can be. Section 7 takes this perspective, offering two lower bounds on session throughput. An upper bound on the cross-network delay of each packet is also provided; this delay bound is proportional to the window size, of course. Putting the pieces together, we see that round-robin scheduling affords certain baseline throughput guarantees regardless of window size, and window size selection then involves a tradeoff between improving throughput fairness and reducing packet delay.

Section 8 summarizes this research, discusses its application to networks with diverse user requirements, and offers

1. It is known that without round-robin link scheduling, things can be even worse: Reference [41] shows that if first-come-first-served link scheduling is used, then even large windows cannot guarantee throughput fairness.

suggestions for further study.

2. SYSTEM MODEL

This section presents a system model featuring uniform link capacities, uniform packet lengths, a fixed set of sessions, virtual circuit routing, link-by-link window flow control, and round-robin link scheduling. Some assumptions detailed below (e.g., noiseless, reliable links without propagation delays) are clearly unrealistic, but they allow us to focus on the simplest version of the problem.

2.1 Nodes, Links, Packets, and Time

The network consists of store-and-forward nodes joined by point-to-point bidirectional links. (In our figures, nodes are depicted as dots and links as solid lines.) Links are error-free and perfectly reliable. Nodes, too, are perfectly reliable, and the storage capacity of each node is infinite.

Data packets experience only transmission and queueing delays; i.e., there are no processing delays at nodes and no propagation delays over links. All packets have unit length, all links have unit capacity (in each direction), and the packet transmission slots at all links are synchronized, so that the entire network operates with slotted time. Wherever possible, a discrete-time system model will be used in which the t^{th} discrete-time instant, called *time* t, refers to the *end* of the t^{th} time slot. The model begins at time 0. Let $[s, t]$ denote the interval from the beginning of slot s to the end of slot t; if $s > t$, then $[s, t]$ is null. Define $(s, t]$ as $[s+1, t]$, $[s, t)$ as $[s, t-1]$, and (s, t) as $[s+1, t-1]$.

2.2 Routing, Flow Control, and Demand

The network supports one-way communication sessions. Each session transmits its packets over a fixed path through the network. (In our figures, a session is indicated by a dashed line adjacent to its path.) The set of sessions using the network is fixed; i.e., during the period of the analysis, no existing sessions are terminated and no new sessions are initiated. Sessions may already have packets in transit at time 0. Let **S** denote the number of sessions using the network. Let **N** be the maximum number of sessions sharing any link in the network, and let **N(x)** denote the maximum number of sessions using any link in the path of session x. Let **H** be the maximum number of links in the path of any session, and let **H(x)** denote the number of links in the path of session x. *Hop* h of session x, $1 \le h \le H(x)$, is defined as the

h^{th} link in the path of x, including the related functions in the node at the input end of that link. To streamline the analysis, packet arrivals and departures are modeled as transmissions over dummy hops. The session's source is *hop* 0. The session's sink is *hop* $H(x)+1$.

Packets waiting to be transmitted over hop h of session x are said to be in *buffer* h. The number of packets in this buffer at time t is called the *buffer level* $\mathbf{B(x, h, t)}$. The capacity of this buffer (measured in packets) is called the *window size* $\mathbf{W(x, h)}$. We make the following assumptions about the window sizes: [2]

$$W(x,0) = \infty , \quad 1 \le W(x,1) \le \infty , \quad 1 \le W(x,h) < \infty \text{ for } 2 \le h \le H(x) , \quad 2 \le W(x,H(x)+1) < \infty$$

(Throughout most of this paper, we will assume that the window size $W(x, h)$ is constant for all sessions x and for hops $1 \le h \le H(x)$.)

The following restriction prevents a session x from overflowing any buffer $h > 0$: if buffer h is full at the end of slot $t-1$, the *window* is said to be *closed*, and session x may not transmit a packet over hop $h-1$ during slot t. In order that hop $h-1$ know the level of buffer h, hop h sends a notice upstream to hop $h-1$ during a time slot if a packet for session x is being transmitted over hop h during that slot. This notice requires few bits and can be piggybacked onto a data packet if any are available. Therefore the link capacity consumed by these notices will be ignored.

Packet arrivals for a session x are modeled as follows. Buffer 0 contains an infinite supply of packets that have not yet arrived at the network. During occasional time slots, the session source (hop 0) tries to transfer a packet from buffer 0 to buffer 1. If the window for buffer 1 is closed, then the attempt fails and the packet transfer does not take place. The number of such attempts (called *chances at hop* 0) during interval $(s, t]$ is denoted by $\mathbf{C(x, 0, s, t)}$. For $s \ge t$, $C(x, 0, s, t)$ is defined to be zero. The *demand rate* $\lambda(\mathbf{x})$ is roughly defined as the frequency of these attempts, measured in chances per slot.

Packet departures for a session x are modeled as follows. The session's sink (hop $H(x)+1$) absorbs a packet from buffer $H(x)+1$ during the slot immediately following the packet's transmission over hop $H(x)$. Hence buffer $H(x)+1$ never contains more than one packet and is never full. [3]

---

2. A session with high demand and no competition needs window sizes of at least two to achieve free flow.

3. Fast session sinks have been assumed for simplicity. Our analysis can easily be generalized to a sink model similar to the source model.

The throughput $\mathbf{P(x, h, s, t)}$ is defined as the number of packets transmitted by session x over hop h during interval $(s, t]$. For $s \geq t$, $P(x, h, s, t)$ is defined to be zero. If link l is hop h for session x, then $\mathbf{P'(x, l, s, t)}$ is defined to equal $P(x, h, s, t)$. The long-term average throughput $\mathbf{R_A(x)}$ of a session x is defined as the limit (if it exists) of $P(x, H(x), 0, t) / t$ as t tends to infinity.

2.3 Scheduling

The scheduler for each link l considers the sessions using l as elements of a directed ring. At the beginning of each time slot t, the scheduler searches the ring, starting immediately *after* the session x that used l most recently, until the first session with a waiting packet and an open window is found. If there is such a session, it transmits a packet over l during slot t. If the ring is searched through session x without success, then the search stops after x, and nothing is transmitted. Each session examined in this search is said to have been offered one *chance* to use link l. Let $\mathbf{C'(x, l, s, t)}$ denote the number of chances offered to session x at link l during interval $(s, t]$. If link l is hop h for session x, then $\mathbf{C(x, h, s, t)}$ is defined to equal $C'(x, l, s, t)$. For $s \geq t$, define $C'(x, l, s, t)$ and $C(x, h, s, t)$ to be zero.

3. FAIRNESS CRITERION

This section describes the max-min flow criterion, our definition of throughput fairness. The version presented here was developed by Hayden [9]. Similar versions were proposed independently by Jaffe [20, 21] and Luss and Smith [25]. The criterion is described here as it applies to the system model presented in Section 2. In particular, it is assumed that all links have unit capacity (in each direction) and that the sessions and their routes and their demand rates have been specified.

First, let us define some terms. An *allocation* r is a function that assigns each session a rate consistent with its demand and the link capacities; i.e., for each session x, $0 \leq r(x) \leq \lambda(x)$, and for each link l, $\sum_{x \text{ using } l} r(x) \leq 1$. A session x is *satisfied* if its assigned rate equals its demand rate. A link l is a *bottleneck* for a session x using l if x's assigned rate is at least as large as that of any other session using l, and if the entire capacity of l is assigned to the sessions using it. The *rate list* of an allocation r is a unique vector consisting of the rates $r(x)$ assigned to all the sessions x. If the same rate is assigned to k different sessions, then that rate appears k times in the rate list. The

components of the rate list must appear in nondecreasing order.

Now fairness can be defined. An allocation r satisfies the *max-min flow criterion* if no other allocation has a rate list that is lexicographically greater than the rate list of r. Roughly speaking, this means that the smallest rate assigned to any session by r is as large as possible and, subject to that constraint, the second-smallest assigned rate is as large as possible, etc. Each of these nested optimization problems can be formulated as a linear program [9], and it is not difficult to show that there exists a unique allocation that solves them all. The rate assigned by this unique max-min allocation to a session x is called its *fair rate* $\mathbf{R_F(x)}$. The *congestion index* $\mathbf{I(x)}$ of a session x tells how its fair rate compares with the fair rates of the other sessions. All sessions with the smallest fair rate have congestion index 1, all sessions with the second-smallest fair rate have congestion index 2, etc.

It is not difficult to prove that an allocation satisfies the max-min flow criterion if and only if every unsatisfied session has at least one bottleneck link. Once this is established, it is easy to see why round-robin link scheduling might be expected to achieve the max-min fair rates. Consider a session whose demand exceeds its throughput. Packets for this session should accumulate at the input to its most congested link. Therefore, the session should never have to forfeit its turn in that link's round robin. This ensures that the session's average throughput will be at least as large as that of any of its competitors at that link, and it also ensures that the link will stay busy. Thus the link should be a bottleneck link for that session in the technical sense defined above. Every session that is not limited by its own demand should have such a bottleneck link; hence the resulting average throughputs should equal the max-min fair rates. Of course, this crude plausibility argument does not constitute a proof. [4]

For the remainder of this paper, the term *bottleneck* will mean *bottleneck with respect to the max-min allocation*. In other words, link l is a bottleneck for a session x using l if $R_F(x) \geq R_F(y)$ for all sessions y using l and if $\sum_{y \text{ using } l} R_F(y) = 1$. Hop 0 is said to be a bottleneck for x if x is satisfied, i.e., if $R_F(x) = \lambda(x)$. Every session x has at least one bottleneck hop h in the range $0 \leq h \leq H(x)$. (Hop $H(x)+1$ is never said to be a bottleneck hop.)

4. The biggest hole in the argument above is that window flow control might impede a session's transmission over a bottleneck. To avoid such interference, the window size for a session x should be large enough to compensate for fluctuations in the bandwidth proffered to x by the schedulers of its links. Unfortunately, a large window permits large fluctuations in x's flow, which will occur at least initially as queues build upstream of x's bottleneck(s). Session x's flow variations themselves contribute to the fluctuations in the polling processes at x's links, thereby disturbing the flows of x's competitors and boosting their window size requirements, which makes their flows still burstier and affects *their* competitors, etc. Because of these complex interactions, the system performance is not obvious. The goal of this paper, therefore, is to provide a rigorous performance analysis.

We conclude this section with the example of Figure 1. Every session has a demand rate of 1, except session x_4, whose demand rate is 1/6, and session x_6, whose demand rate is 1/3. The max-min fair rate for each session is 1/3, except session x_4, whose fair rate is 1/6, and session x_5, whose fair rate is 1/2. Session x_4 has congestion index 1. Session x_5 has congestion index 3. The other sessions have congestion index 2. The rate list for the max-min allocation is (1/6, 1/3, 1/3, 1/3, 1/3, 1/3, 1/2). Sessions x_1 and x_2 have link l_1 as a bottleneck. Session x_3 has two bottleneck links -- l_1 and l_4. The bottleneck hop for x_4 is hop 0 -- its demand. Link l_3 is the bottleneck for x_5. Session x_6 has its demand and l_4 as bottlenecks. Link l_4 is also the bottleneck for x_7. Link l_2 is not a bottleneck for any session, since it has unused capacity.

4. TRANSIENT ANALYSIS FOR SMOOTH DEMAND

This section analyzes a system during an interval (T_1, T_2) of smooth demand. Specifically, it is assumed that there exists a constant Δ such that the demand of each session x over each subinterval $(s, t]$ of (T_1, T_2) is within Δ packets of the nominal amount $\lambda(x)\cdot(t-s)$. Special cases of this demand model include: (1) evenly spaced arrivals of one packet every $1/\lambda(x)$ slots ($\Delta = 1$), (2) packet arrivals evenly spaced by $1/\lambda(x)$ slots except for a jitter of $\pm j$ slots ($\Delta = 2j\lambda(x)+1$), (3) roughly periodic arrivals of k packets within each successive block of $k/\lambda(x)$ slots ($\Delta = 2k$), and (4) demand so heavy that packets are always waiting to enter the network (which can be modeled by $\lambda(x) = 1$ and $\Delta = 0$, regardless of the actual mean and variation of the demand). Large windows are also assumed. Theorem 1 concludes that the throughput of each session x at each hop over each subinterval $(s, t]$ of (T_1, T_2) is within a constant number of packets of the fair amount $R_F(x)\cdot(t-s)$. This constant is a function of Δ and the window size, but is independent of the length $(t-s)$ of the subinterval. This dependence on the window size is not surprising, since the system should go through an unfair transient period during which buffers upstream of bottleneck hops fill and buffers downstream of bottleneck hops drain.

Before formally stating Theorem 1, let us state a lemma on which it depends. Lemma 1 gives a lower bound for the packet flow of a session x, given lower bounds for x's demand and for the service offered to x by any round-robin link scheduler. Link-by-link windows make this a problem of tandem queues with large but finite buffers. Very roughly speaking, Lemma 1 claims that x's throughput rate equals the minimum of its demand rate and the link service rates, with short-term throughput variations not much worse than the sum of the variations of the demand and service processes.

Lemma 1 is proved in the Appendix.

Lemma 1

Let x be some session. Let T_1 and T_2 be times satisfying $0 \le T_1 < T_2 \le \infty$. (Note that T_2 is permitted to be infinite.) Suppose there exist real numbers r, e and f such that the following inequality holds for every hop h of x in the range $0 \le h \le H(x)$, for any positive integer K, and for any nondecreasing sequence $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ of times in $[T_1, T_2)$:

(1) $$\sum_{k=1}^{K} C(x, h, s_k, t_k) \ge r \cdot \sum_{k=1}^{K} (t_k - s_k) - e - K \cdot f$$

Suppose that buffers 1 through $H(x)$ of session x have the same capacity W, and

(2) $$[H(x) + 1] \cdot (f + 1) \le W < \infty$$

It follows that

(3) For each hop h of x in the range $0 \le h \le H(x)$, for any positive integer K, and for any nondecreasing sequence $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ of times in $[T_1, T_2)$,
$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \ge r \cdot \sum_{k=1}^{K} (t_k - s_k) - [H(x) + 1] \cdot e - K \cdot \left[[H(x) + 1] \cdot f + H(x) \right]$$

Theorem 1

Let T_1 and T_2 be times satisfying $0 \le T_1 < T_2 \le \infty$. (Note that T_2 is permitted to be infinite.) Suppose there exists a nonnegative real number Δ such that, for each session x and for all times s and t satisfying $T_1 \le s \le t < T_2$,

(4) $$\left| C(x, 0, s, t) - \lambda(x) \cdot (t - s) \right| \le \Delta$$

Suppose that buffers 1 through $H(x)$ of all sessions x have the same capacity W, and

(5) $$3 \cdot (H+1)^S \cdot N^{S-1} \cdot (\Delta+2) \le W < \infty$$

It follows that, for each session x, for each hop h of x in the range $0 \le h \le H(x)$, and for all times s and t satisfying $T_1 \le s \le t < T_2$,

(6) $$\left| P(x, h, s, t) - R_F(x)\cdot(t - s) \right| \le (H+1)^S \cdot N^{2S-1} \cdot (W + 3\cdot\Delta + 4)$$

Proof of Theorem 1

Define the following functions of integers $i \ge 1$ and $K \ge 0$. (The subscripts C, P, L, and U below stand for *chances*, *packets*, *lower bound*, and *upper bound*, respectively.)

$$\eta(i) = \begin{cases} 0 & \text{for } i = 1 \\ 1 & \text{for } i > 1 \end{cases}$$

$$D_{CL}(i, K) = \eta(i)\cdot(H + 1)^{i-1}\cdot(N - 1)\cdot N^{2i-3}\cdot(W + 2\Delta + 2) + K\cdot[(H + 1)^{i-1}\cdot N^{i-1}\cdot(\Delta + 2) - 1]$$

$$D_{PL}(i, K) = \eta(i)\cdot(H + 1)^{i} \cdot (N - 1)\cdot N^{2i-3}\cdot(W + 2\Delta + 2) + K\cdot[(H + 1)^{i} \cdot N^{i-1}\cdot(\Delta + 2) - 1]$$

$$D_{PU}(i, K) = (H + 1)^{i} \cdot N^{2i-1} \cdot (W + 2\Delta + 2) + K\cdot(H + 1)^{i} \cdot N^{i-1}\cdot(\Delta + 2)$$

In order to show (6), properties (7) - (9) will be proved.

(7) For each session x, for each hop h of x in the range $0 \le h \le H(x)$, for any positive integer K,
and for any nondecreasing sequence $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ of times in $[T_1, T_2)$:
$$\sum_{k=1}^{K} C(x, h, s_k, t_k) \ge R_F(x)\cdot\sum_{k=1}^{K}(t_k - s_k) - D_{CL}(I(x), K)$$

(8) For each session x, for each hop h of x in the range $0 \le h \le H(x)$, for any positive integer K,
and for any nondecreasing sequence $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ of times in $[T_1, T_2)$:
$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \ge R_F(x)\cdot\sum_{k=1}^{K}(t_k - s_k) - D_{PL}(I(x), K)$$

(9) For each session x, for each hop h of x in the range $0 \le h \le H(x)$, for any positive integer K,
and for any nondecreasing sequence $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ of times in $[T_1, T_2)$:
$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \le R_F(x)\cdot\sum_{k=1}^{K}(t_k - s_k) + D_{PU}(I(x), K)$$

The proof is by induction on the congestion index $I(x)$ of the session x. Very roughly speaking, the proof constructs the performance bounds as follows:

$$D_{CL}(i, \cdot) \sim N \cdot D_{PU}(i-1, \cdot)$$
$$D_{PL}(i, \cdot) \sim H \cdot D_{CL}(i, \cdot)$$
$$D_{PU}(i, \cdot) \sim N \cdot D_{PL}(i, \cdot)$$

Only the proof of the induction step will be presented, because the proof of the base case (viz., $I(x) = 1$) is nearly identical. Fix a congestion index $i > 1$. The induction hypothesis asserts that (7), (8), and (9) hold for all sessions x with $I(x) < i$. It must be shown that (7), (8), and (9) hold for all sessions x with $I(x) = i$.

First consider (7). Let x be any session with $I(x) = i$. Let h be any hop of x in the range $0 \le h \le H(x)$. Let K be any positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be any nondecreasing sequence of times in $[T_1, T_2)$. For $h = 0$, the desired result follows from assumption (4):

$$\sum_{k=1}^{K} C(x, 0, s_k, t_k) \quad \ge \quad \lambda(x) \cdot \sum_{k=1}^{K} (t_k - s_k) - K \cdot \Delta \quad \ge \quad R_F(x) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{CL}(i, K)$$

For $1 \le h \le H(x)$, the properties of round-robin scheduling must be used. Let l denote the link corresponding to hop h of x. Let Y be the (possibly empty) set of sessions y on l for which $I(y) < i$. Let Z be the set of sessions z on l for which $I(z) \ge i$. Note that Z includes x. For $k = 1, 2, \ldots, K$, let q_k be the number of slots in $(s_k, t_k]$ that are not used by sessions in Y:

$$q_k = (t_k - s_k) - \sum_{y \in Y} P'(y, l, s_k, t_k) \tag{10}$$

Since these q_k slots are not used by Y, the round-robin scheduler at l will offer each of these slots to at least one session in Z; hence

$$q_k \quad \le \quad \sum_{z \in Z} C'(z, l, s_k, t_k) \quad \le \quad |Z| \cdot \max_{z \in Z} C'(z, l, s_k, t_k) \tag{11}$$

By the operating rules of the round-robin scheduler, session x must receive almost as many chances as any other session on l during $(s_k, t_k]$; in particular,

$$C'(x, l, s_k, t_k) \ge \max_{z \in Z} C'(z, l, s_k, t_k) - 1 \tag{12}$$

Combining (12), (11) and (10) and summing over k yields:

$$\sum_{k=1}^{K} C'(x, l, s_k, t_k) \ge \frac{1}{|Z|} \cdot \left\lceil \sum_{k=1}^{K} (t_k - s_k) - \sum_{y \in Y} \sum_{k=1}^{K} P'(y, l, s_k, t_k) \right\rceil - K$$

By the induction hypothesis, (9) holds for all sessions y in Y; hence

$$\sum_{k=1}^{K} C'(x, l, s_k, t_k) \geq \frac{1}{|Z|} \cdot \left\lceil \sum_{k=1}^{K} (t_k - s_k) - \sum_{y \in Y} \left\lceil R_F(y) \cdot \sum_{k=1}^{K} (t_k - s_k) + D_{PU}(I(y), K) \right\rceil \right\rceil - K$$

$$= \frac{1}{|Z|} \cdot \left[1 - \sum_{y \in Y} R_F(y) \right] \cdot \sum_{k=1}^{K} (t_k - s_k) - \frac{1}{|Z|} \cdot \sum_{y \in Y} D_{PU}(I(y), K) - K$$

$$\geq \frac{1}{|Z|} \cdot \left[\sum_{z \in Z} R_F(z) \right] \cdot \sum_{k=1}^{K} (t_k - s_k) - \frac{|Y|}{|Z|} \cdot D_{PU}(i-1, K) - K$$

$$\geq \frac{1}{|Z|} \cdot \left[\sum_{z \in Z} R_F(x) \right] \cdot \sum_{k=1}^{K} (t_k - s_k) - (N - 1) \cdot D_{PU}(i-1, K) - K$$

$$\geq R_F(x) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{CL}(i, K)$$

This completes the proof of (7).

Now (8) will be proved, using the assumption of large flow control windows. Let x be any session with $I(x) = i$. Lemma 1 will be invoked, with $r = R_F(x)$, $e = (H + 1)^{i-1} \cdot (N - 1) \cdot N^{2i-3} \cdot (W + 2\Delta + 2)$, and $f = (H + 1)^{i-1} \cdot N^{i-1} \cdot (\Delta + 2) - 1$. Condition (1) of Lemma 1 holds because (7) was just proved for all sessions with congestion index i. Condition (2) of Lemma 1 is satisfied by assumption (5). Conclusion (3) of Lemma 1 asserts that for each hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for any nondecreasing sequence $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ of times in $[T_1, T_2)$:

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq R_F(x) \cdot \sum_{k=1}^{K} (t_k - s_k) - (H + 1) \cdot D_{CL}(i, K) - K \cdot H \geq R_F(x) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{PL}(i, K)$$

This completes the proof of (8).

Next (9) will be proved, using the properties of the max-min fair rates and the assumption of window flow control. Let x be any session with $I(x) = i$. Let h be any hop of x in the range $0 \leq h \leq H(x)$. Let K be any positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be any nondecreasing sequence of times in $[T_1, T_2)$. Consider the following claim:

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) = P(x, h, s_1, t_K) - \sum_{k=1}^{K-1} P(x, h, t_k, s_{k+1})$$

$$\leq P(x, h, s_1, t_K) - R_F(x) \cdot \sum_{k=1}^{K-1} (s_{k+1} - t_k) + D_{PL}(i, K-1) \tag{13}$$

If $K = 1$, then (13) holds because $D_{PL}(i, 0) \geq 0$. If $K > 1$, then (13) holds because (8) was just proved for all sessions with congestion index i. Now the term $P(x, h, s_1, t_K)$ in (13) will be bounded. Let h^* be any bottleneck hop of x, $0 \leq h^* \leq H(x)$. (Recall that the max-min flow criterion guarantees the existence of h^*.) Since x is subject to link-by-link window flow control, and x has at most H buffers between hops h and h^*, and each buffer has capacity W,

$$P(x, h, s_1, t_K) \leq P(x, h^*, s_1, t_K) + W \cdot H \tag{14}$$

Next the term $P(x, h^*, s_1, t_K)$ in (14) will be analyzed, using the properties of bottleneck hops. There are two cases to consider. If $h^* = 0$, it follows from assumption (4) and the definition of a bottleneck that

$$P(x, 0, s_1, t_K) \leq C(x, 0, s_1, t_K) \leq \lambda(x) \cdot (t_K - s_1) + \Delta = R_F(x) \cdot (t_K - s_1) + \Delta \tag{15}$$

If $1 \leq h^* \leq H(x)$, let l denote the link corresponding to hop h^*, and let Y denote the set of sessions $y \neq x$ that use l. Obviously, x can only use slots in $(s_1, t_K]$ that are not used by sessions in Y:

$$P(x, h^*, s_1, t_K) \leq (t_K - s_1) - \sum_{y \in Y} P'(y, l, s_1, t_K)$$

Note that, by the definition of a bottleneck, $I(y) \leq I(x)$ for all sessions y in Y. For those sessions y in Y with $I(y) < i$, the induction hypothesis permits the application of (8) above. Furthermore, since (8) was just proved for all sessions with congestion index i, (8) applies to the other sessions y in Y as well. By (8) and the definition of a bottleneck,

$$P(x, h^*, s_1, t_K) \leq (t_K - s_1) - \sum_{y \in Y} \left[R_F(y) \cdot (t_K - s_1) - D_{PL}(I(y), 1) \right]$$

$$\leq \left[1 - \sum_{y \in Y} R_F(y) \right] \cdot (t_K - s_1) + |Y| \cdot D_{PL}(i, 1)$$

$$\leq R_F(x) \cdot (t_K - s_1) + (N - 1) \cdot D_{PL}(i, 1) \tag{16}$$

Inequalities (15) (for $h^* = 0$) and (16) (for $h^* > 0$) may be combined into a single inequality:

(17) $$P(x, h^*, s_1, t_K) \le R_F(x)\cdot(t_K - s_1) + \Delta + (N-1)\cdot D_{PL}(i, 1)$$

Substituting (17) into (14) and (14) into (13) yields:

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \le R_F(x)\cdot\sum_{k=1}^{K}(t_k - s_k) + (N-1)\cdot D_{PL}(i, 1) + D_{PL}(i, K-1) + W\cdot H + \Delta$$

$$\le R_F(x)\cdot\sum_{k=1}^{K}(t_k - s_k) + D_{PU}(i, K)$$

This completes the proof of (9).

Conclusion (6) follows from results (8) - (9) and the fact that $I(x) \le S$ for all sessions x.

□

5. STEADY-STATE ANALYSIS FOR SMOOTH DEMAND

This section examines the steady-state behavior of systems with smooth demand. The demand assumptions are the same as for Section 4, but now they apply for all time. Large windows are again assumed. Theorem 2 concludes that the long-term average throughput $R_A(x)$ of each session x equals its fair rate $R_F(x)$. In other words, smooth demand and large windows are *sufficient* for throughput fairness. (An example is included to show that large windows are sometimes *necessary* as well, and that throughputs can be very unfair if the windows are too small.) Theorem 2 also offers the following steady-state analog of Theorem 1: there exists a time $T_{SS} \ge 0$ such that the throughput of each session x at each hop over each interval $(s, t]$ later than T_{SS} is within a constant number of packets of the fair amount $R_F(x)\cdot(t-s)$. This constant is a function of Δ, but is independent of the length $(t-s)$ of the interval. Moreover, the constant is independent of the window size (in contrast with Theorem 1). Several corollaries of Theorem 2 relating the steady-state buffer levels to the locations of the bottleneck hops can be found in [41] and [1].

Theorem 2

Suppose there exists a nonnegative real number Δ such that $| C(x, 0, s, t) - \lambda(x)\cdot(t - s) | \le \Delta$ for each session x and for all times s and t satisfying $0 \le s \le t$. Also suppose that buffers 1 through $H(x)$ of all sessions x have the same capacity W, and that $3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2) \le W < \infty$. It follows that the long-term average throughput $R_A(x)$

exists for each session x and equals the fair rate $R_F(x)$. Furthermore, there exists a time $T_{SS} \geq 0$ such that, for each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS} \leq s \leq t$,

(18) $$\left| P(x, h, s, t) - R_F(x) \cdot (t - s) \right| \leq (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2)$$

Sketch of Proof of Theorem 2

The existence and fairness of the long-term average throughputs are consequences of Theorem 1. Let I denote the number of distinct fair rates in the network, i.e., the maximum congestion index of any session. It will be shown that there exist times $T_{SS}(0)$, $T_{SS}(1)$, ..., $T_{SS}(I)$ such that $0 = T_{SS}(0) \leq T_{SS}(1) \leq \cdots \leq T_{SS}(I)$ and such that the following properties hold.

(19) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS}(I(x)-1) \leq s \leq t$:

$$C(x, h, s, t) \geq R_F(x) \cdot (t - s) - (H + 1)^{I(x)-1} \cdot N^{I(x)-1} \cdot (\Delta + 2) + 1$$

$$P(x, h, s, t) \geq R_F(x) \cdot (t - s) - (H + 1)^{I(x)} \cdot N^{I(x)-1} \cdot (\Delta + 2) + 1$$

$$P(x, h, s, t) \leq R_F(x) \cdot (t - s) + (H + 1)^{I(x)} \cdot N^{I(x)} \cdot (W + 2\Delta + 2)$$

(20) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS}(I(x)) \leq s \leq t$:

$$P(x, h, s, t) \leq R_F(x) \cdot (t - s) + (H + 1)^{I(x)} \cdot N^{I(x)-1} \cdot (\Delta + 2)$$

The proof is by induction on the congestion index $I(x)$ of the session x. Only the proof of the induction step will be discussed, because the proof of the base case (viz., $I(x) = 1$) is nearly identical. Fix a congestion index $i > 1$. The induction hypothesis asserts that there exist times $T_{SS}(0)$, $T_{SS}(1)$, ..., $T_{SS}(i-1)$ such that $0 = T_{SS}(0) \leq T_{SS}(1) \leq \cdots \leq T_{SS}(i-1)$ and such that properties (19) and (20) hold for all sessions x with $I(x) < i$. It must be shown that, for such a time $T_{SS}(i-1)$, property (19) also holds for all sessions x with $I(x) = i$. It must also be shown that there exists a time $T_{SS}(i) \geq T_{SS}(i-1)$ such that property (20) holds for all sessions x with $I(x) = i$. The proof of (19) in the induction step is similar to the proofs of (7) - (9) and can be found in [41]. Only simple algebra must be applied to (19) to deduce the existence of $T_{SS}(i)$ satisfying (20); a straightforward proof by contradiction appears in [41]. Once (19) and (20) have been established, conclusion (18) follows by taking T_{SS} to be $T_{SS}(I)$ and noting that $I(x) \leq S$.

□

Example

This example shows how lots of sessions with small fair rates can conspire to keep another session from achieving its larger fair rate. Consider a system with the layout shown in Figure 2. Every session has heavy demand, so the max-min fair rate for session x is $½$, while the other sessions deserve rates of $1/N$ each. The window size W is in the range $2 \le W < ½N$. Divide time into intervals of length N, viz., $[1, N]$, $[N+1, 2N]$, ... The round-robin rings and initial conditions are such that the system is periodic with period N, sessions $y_1, \cdots, y_{½N}$ each transmit one packet over link l_1 in the first half of each interval, and sessions $z_1, \cdots, z_{½N}$ each transmit one packet over link l_2 in the second half of each interval. Initially, session x has W packets waiting for transmission over link l_2. During the first half of each interval, session x transmits no packets over link l_1 (because the link is busy serving other sessions) and only W packets over l_2 (because x runs out of packets). During the second half of each interval, x transmits no packets over l_2 (because the link is busy serving other sessions) and only W packets over l_1 (because of flow control). The long-term average throughput of session x is W/N, which is less than its fair rate of $½$. The long-term average throughputs of the other sessions are fair. A fraction $½ - W/N$ of the capacities of l_1 and l_2 are wasted. The unfairness of this example is due to the unfortunate choice of round-robin rings and initial conditions. It is easy to choose other values that result in fair throughputs even for $W = 2$.

6. STEADY-STATE ANALYSIS FOR BERNOULLI DEMAND

This section assumes that the sessions have independent Bernoulli demand processes. Theorem 3 shows that the session throughput rates can be made arbitrarily close to the fair rates by choosing a sufficiently large window size.

Theorem 3

Suppose that buffers 1 through $H(x)$ of all sessions x have the same finite capacity W. Suppose that the demands of the sessions are independent. For each session x, suppose chances at hop 0 form a Bernoulli process with rate $\lambda(x)$, $0 \le \lambda(x) \le 1$. It follows that, with probability one, the long-term average throughput $R_A(x)$ exists for each session x and

$$(21) \qquad \left| R_A(x) - R_F(x) \right| \leq \frac{74 \cdot S \cdot (H+1)^{2S} \cdot N^{2S-1}}{W^{0.5}}$$

Proof of Theorem 3

Assume that $W \geq 12 \cdot (H+1)^S \cdot N^{S-1}$, since otherwise (21) is obvious. Define a real number Δ as follows.

$$(22) \qquad \Delta = \frac{W}{6 \cdot (H+1)^S \cdot N^{S-1}}$$

Let τ be some integer in the range

$$(23) \qquad W^{1.5} \leq \tau \leq 2 \cdot W^{1.5}$$

Divide the time interval $(0, \infty)$ into non-overlapping subintervals of length τ slots: $(0, \tau]$, $(\tau, 2\tau]$, ... A subinterval $(k \cdot \tau, (k+1) \cdot \tau]$ is said to be *regular* if $\left| C(x, 0, s, t) - \lambda(x) \cdot (t - s) \right| \leq \Delta$ for all sessions x and for all times s and t satisfying $k \cdot \tau \leq s < t \leq (k+1) \cdot \tau$. Let π be the probability that a given subinterval $(k \cdot \tau, (k+1) \cdot \tau]$ is irregular. Note that, by Kolmogorov's inequality [43],

$$(24) \qquad \pi \leq \sum_x \text{PROB} \left\{ \begin{array}{c} \left| C(x, 0, k \cdot \tau, u) - \lambda(x) \cdot (u - k \cdot \tau) \right| > \frac{1}{2}\Delta \\ \text{for some } u \text{ such that } k \cdot \tau < u \leq (k+1) \cdot \tau \end{array} \right\} \leq \sum_x \frac{\lambda(x) \cdot [1 - \lambda(x)] \cdot \tau}{(\frac{1}{2}\Delta)^2} \leq \frac{S \cdot \tau}{\Delta^2}$$

For every positive integer K, let q_K denote the number of irregular subintervals among $(0, \tau]$, $(\tau, 2\tau]$, ... , $((K-1) \cdot \tau, K \cdot \tau]$.

Theorem 1 gives the following throughput bound for any regular subinterval:

$$\left| P(x, H(x), (k-1) \cdot \tau, k \cdot \tau) - R_F(x) \cdot \tau \right| \leq (H+1)^S \cdot N^{2S-1} \cdot (W + 3 \cdot \Delta + 4) \leq 2 \cdot W \cdot (H+1)^S \cdot N^{2S-1}$$

For irregular subintervals, we can only say that

$$\left| P(x, H(x), (k-1) \cdot \tau, k \cdot \tau) - R_F(x) \cdot \tau \right| \leq \tau$$

Combining these bounds in the correct proportions gives the following bound for any positive integer K:

$$\left| P(x, H(x), 0, K \cdot \tau) - R_F(x) \cdot K \cdot \tau \right| \leq q_K \cdot \tau + (K - q_K) \cdot [2 \cdot W \cdot (H+1)^S \cdot N^{2S-1}]$$

Divide by $K \cdot \tau$ and let K tend to infinity. The long-term average throughput $R_A(x)$ almost surely exists, since the system is a finite Markov chain [44]. Moreover, the frequency of irregular subintervals converges almost surely to π, by the strong law of large numbers [43]. Therefore, with probability one,

$$| R_A(x) - R_F(x) | \leq \pi + \frac{2 \cdot W \cdot (H + 1)^S \cdot N^{2S-1}}{\tau}$$

Applying (24), (23), and (22) gives the desired result (21).

□

7. THROUGHPUT AND DELAY GUARANTEES WITHOUT LARGE WINDOWS

This section considers a particular session x whose window sizes for hops 2 through $H(x)+1$ are only assumed to be at least two. The window sizes of the other sessions in the network are arbitrary (i.e., these window sizes only need to satisfy the basic assumptions of Section 2.2). Some assumptions concerning x's demand are made, but the demands of the other sessions in the network are arbitrary (not even well-defined demand rates are assumed). Clearly, the conditions of this section are much less restrictive than those of earlier sections; hence the results of this section are probably of greater practical value. Two lower bounds on x's throughput are given, as well as an upper bound on the cross-network delay of each packet of x. Proofs of these bounds may be found in [41].

Theorem 4

Suppose that the window sizes for hops 1 through $H(x)+1$ of some session x are at least 2 but finite. Suppose that chances for session x at hop 0 form a Bernoulli process with rate $\lambda(x)$, and $0 < \lambda(x) \leq 1$. Suppose that the demand of session x is independent of the demands of the other sessions in the network. If the demand processes of the other sessions are not well-behaved, then the long-term average throughput $R_A(x)$ may not exist. However, x's throughput $P(x, H(x), 0, t)$ *is* bounded below by a stochastic process $\Phi(t)$ whose long-term average rate (with probability one) exists and equals $\frac{\lambda(x)}{[1 - \lambda(x)]^{N(x)} + N(x) \cdot \lambda(x)}$.

As one would expect, this guaranteed rate tends to zero as $\lambda(x)$ tends to zero or as $N(x)$ tends to infinity. As $\lambda(x)$ tends to one, the guaranteed rate tends to $1/N(x)$, and as $N(x)$ tends to one, the guaranteed rate tends to $\lambda(x)$; these limits are also intuitive.

Theorem 5

This theorem shows that the guaranteed rate can be increased by allowing a session to buffer more of its demand. Suppose that buffer 1 of some session x has infinite capacity and that the window sizes for hops 2 through $H(x)+1$ are at least 2 but finite. Suppose that the times between chances for session x at hop 0 are independent and identically distributed, with mean $1/\lambda(x)$, and $0 < \lambda(x) \leq 1$. (Because buffer 1 is infinite, each such chance will result in a packet transmission.) If the demand processes of the other sessions are not well-behaved, then the long-term average throughput $R_A(x)$ may not exist. However, x's throughput $P(x, H(x), 0, t)$ *is* bounded below by a stochastic process $\Phi(t)$ whose long-term average rate (with probability one) exists and equals $\min\left\lceil \frac{1}{N(x)}, \lambda(x) \right\rceil$.

How does this guaranteed rate compare with the max-min fair rate? Suppose that the system's demand processes are so regular that the demand rates $\lambda(y)$ and the long-term average throughputs $R_A(y)$ happen to exist for all sessions y. Then the max-min fair rates $R_F(I(y))$ also exist, and the long-term average throughput for session x is guaranteed by Theorem 5 to be within a factor of $N(x)$ of its fair rate. The example in Section 5 shows that unfairness factors proportional to $N(x)$ are actually possible. [5]

Theorem 6

Suppose that the window sizes for hops 2 through $H(x)+1$ of some session x are at least 2 but finite. The capacity of buffer 1 is arbitrary, possibly even infinite. The demands of all sessions, including session x, are arbitrary. It follows that, for any packet p of session x that enters buffer 1, the cross-network delay of p is at most $\left\lceil \sum_{h=2}^{H(x)} W(x, h) \right\rceil \cdot N(x) + 1$. This delay includes the transmission delays across hops 1 through $H(x)$, plus the queuing delays in buffers 2 through $H(x)$.

If the system is sufficiently well-behaved that the average throughput and the average cross-network delay exist for session x, then the delay guarantee of Theorem 6 can be compared with that found by applying Little's formula [45] to

5. The long-term average session throughput for this example is the same whether $W(x, 1)$ is finite or infinite.

the collection of buffers 2 through $H(x)$. The latter approach gives the following upper bound on the *average* cross-network delay per packet: $\left\lceil \sum_{h=2}^{H(x)} W(x, h) \right\rceil \cdot \frac{1}{R_A(x)} + 1$. This bound may be tighter than the bound of Theorem 6 in some cases. Note, however, that Theorem 6's bound applies to *each* packet of session x.

8. CONCLUSIONS

Round-robin scheduling with windows is a simple technique for allocating link capacity among competing sessions in a packet network. If a sufficiently large window size is used throughout the network, the session throughput rates can be made arbitrarily close to the ideal max-min fair rates. This performance is suited to applications such as large file transfers. Other applications may not be able to take advantage of the potentially variable throughput rate offered by the round-robin schedulers or may not be able to tolerate the potentially large cross-network packet delay permitted by the large windows. Fortunately, the round-robin method guarantees a certain minimum service rate to each session for any reasonable window size, even if different sessions use different window sizes. This service rate determines a maximum session throughput rate that can be supported and also roughly determines, for a given throughput rate, the delay of packets waiting to be admitted to the network. The round-robin method also guarantees an upper bound on the cross-network delay for each packet of a session. This bound is proportional to the window size used by that session and is independent of the window sizes of the other sessions in the network. These various results suggest that a network with diverse user requirements might be managed in the following manner. First consider high-bandwidth applications. A session that needs a very large throughput rate at all times should be visited more than once in each round-robin polling cycle, effectively treating it like several standard sessions. (See [31] - [36] for interesting ways to perform round-robin scheduling among non-homogeneous users.) The user would presumably be charged for this privilege. Next consider delay-sensitive applications. A session requiring small packet delays should use a small window to limit the cross-network component of delay. For delay-sensitive sessions with smooth demand, this may be sufficient. However, if the session's packet arrivals are bursty, then long delays may occasionally be incurred as packets wait to be admitted to the network. Some delay-sensitive users may be able to respond effectively to large pre-network queues by compressing their data. A less sophisticated bursty user must purchase a guaranteed service rate substantially larger than its average throughput rate in order to keep pre-network packet delays low. These users should be charged for their high guaranteed service rate, but not as much as if they actually used all their service opportunities. Finally, consider delay-insensitive

applications. Any session that is able to transmit faster than its guaranteed minimum service rate and is willing to tolerate large cross-network delays should use a large window. The results of this paper suggest that the transmission capacity not used by the small-window sessions will be approximately fairly divided among the large-window sessions. These users should not be charged as much for the extra bandwidth that large windows enable them to consume as they would be charged if they reserved that bandwidth by purchasing a larger guaranteed service rate. The flexibility of the large-window users permits more efficient network usage.

This paper focused on the *worst-case* performance of round-robin scheduling with windows. An important area for further study is the *typical* performance of the scheme. Of interest are the following items, as functions of the window size: the fairness of the session throughput rates, the burstiness of the session flows, the severity of transients arising from the initiation and termination of sessions and from changes in session demands, and the cross-network delay. Unfortunately, since many of these performance measures are very sensitive to the network topology, the session routes and demand rates, and the initial conditions, such a study would likely require the simulation of a great many sample networks of at least moderate size. (See [30] for a study of some of these issues.) It would also be worthwhile to examine variations of this method to see if max-min fair throughput rates can be achieved, at least approximately, without computing the rates but without incurring large cross-network delays. One approach is to dynamically adjust the window sizes so that they are no larger than necessary, as in [46] - [55]. (Reference [56] is similar in spirit.) Another possibility is to change the round-robin discipline slightly, e.g., by randomly rearranging the polling order of the sessions from time to time. This might ensure that, even if small windows are used, the system enters very unfair configurations only rarely and only for brief periods.

ACKNOWLEDGMENT

The author is grateful to Prof. Robert G. Gallager for suggesting this topic and supervising this research.

APPENDIX

Lemma 1, which was stated in Section 4, will be proved in this appendix. It depends on Lemma 2, which will be stated and proved first. Lemma 2 derives lower bounds on the packet flow into and out of a buffer during a collection Γ of time slots. The lemma assumes a certain lower bound on the flow into the buffer during subsets of Γ when the buffer is not full. It also assumes a lower bound on the flow out of the buffer when it is not empty. Large windows are assumed as well.

Lemma 2

Let x be some session. Let h be some hop of x in the range $1 \le h \le H(x)$. For convenience, denote $W(x, h)$ by W. Let K be a positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be a nondecreasing sequence of nonnegative times. Assume that the following two properties hold for some real numbers r, e', e'', f', and f'':

(25) If, for $k = 1, 2, \ldots, K$, $J(k)$ is a positive integer, and $v_k^{J(k)}, u_k^{J(k)-1}, v_k^{J(k)-1}, u_k^{J(k)-2}, \ldots, v_k^1, u_k^0$ is a nondecreasing sequence of times in $[s_k, t_k]$ such that $B(x, h, \tau) < W$ for all τ in $\bigcup_{j=1}^{J(k)} [v_k^j, u_k^{j-1})$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^{j-1}) \ge r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - v_k^j) - e' - \left\lceil \sum_{k=1}^{K} J(k) \right\rceil \cdot f'$$

(26) If, for $k = 1, 2, \ldots, K$, $J(k)$ is a positive integer, and $u_k^{J(k)}, v_k^{J(k)}, u_k^{J(k)-1}, v_k^{J(k)-1}, \ldots, u_k^1, v_k^1$ is a nondecreasing sequence of times in $[s_k, t_k]$ such that $B(x, h, \tau) > 0$ for all τ in $\bigcup_{j=1}^{J(k)} [u_k^j, v_k^j)$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^j) \ge r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^j - u_k^j) - e'' - \left\lceil \sum_{k=1}^{K} J(k) \right\rceil \cdot f''$$

Also assume that

(27) $$f' + f'' + 2 \le W < \infty$$

It follows that

(28) $$\sum_{k=1}^{K} P(x, h-1, s_k, t_k) \ge r \cdot \sum_{k=1}^{K} (t_k - s_k) - (e' + e'') - K \cdot (f' + f'' + 1) \qquad \text{and}$$

(29) $$\sum_{k=1}^{K} P(x, h, s_k, t_k) \ge r \cdot \sum_{k=1}^{K} (t_k - s_k) - (e' + e'') - K \cdot (f' + f'' + 1)$$

Proof of Lemma 2

Only the proof of (28) will be presented; the proof of (29) is similar and is found in [41]. For each k, $1 \le k \le K$, let us analyze the time interval $[s_k\ ,\ t_k]$ separately. The first step is to break $[s_k\ ,\ t_k]$ into various subintervals. Determine a positive integer $J(k)$ and define times $u_k^0\ ,\ v_k^1\ ,\ u_k^1\ ,\ v_k^2\ ,\ u_k^2\ ,\ \ldots\ ,\ v_k^{J(k)}\ ,\ u_k^{J(k)}$ by the procedure specified below. An example is shown in Figure 3.

$j \leftarrow 0$

$u_k^0 \leftarrow t_k$

E: $j \leftarrow j + 1$

$v_k^j \leftarrow$ earliest time v in $[s_k\ ,\ u_k^{j-1}]$ that satisfies $B(x, h, \tau) < W$ for all τ in $[v, u_k^{j-1})$

$u_k^j \leftarrow$ earliest time u in $[s_k\ ,\ v_k^j]$ that satisfies $B(x, h, \tau) > 0$ for all τ in $[u, v_k^j)$

if $u_k^j > s_k$ then go to **E**

$J(k) \leftarrow j$

It is not difficult to verify that u_k^j and v_k^j are well-defined and that this procedure terminates. Let us make some remarks about u_k^j and v_k^j :

(30) $s_k = u_k^{J(k)} \le v_k^{J(k)} \le u_k^{J(k)-1} \le v_k^{J(k)-1} \le \cdots \le u_k^1 \le v_k^1 \le u_k^0 = t_k$

(31) $B(x, h, \tau) < W$ for all τ in $[v_k^j\ ,\ u_k^{j-1})\ ,\quad 1 \le j \le J(k)$

(32) $B(x, h, v_k^j) \ge W - 1$ for $1 \le j \le J(k)-1$

(33) If $u_k^{J(k)} < v_k^{J(k)}$, then $B(x, h, v_k^{J(k)}) \ge W - 1$

(34) $B(x, h, \tau) > 0$ for all τ in $[u_k^j\ ,\ v_k^j)\ ,\quad 1 \le j \le J(k)$

(35) $B(x, h, u_k^j) = 1$ for $1 \le j \le J(k)-1$

Now the facts above will be used to analyze the throughput over the subintervals $(u_k^j\ ,\ v_k^j]$. It follows from (32) and (35) that

$$P(x,\, h-1,\, u_k^j\,,\, v_k^j) \;=\; P(x,\, h,\, u_k^j\,,\, v_k^j) \;+ B(x,\, h,\, v_k^j) \;- B(x,\, h,\, u_k^j)$$

$$\geq\; P(x,\, h,\, u_k^j\,,\, v_k^j) \;+ W \,-\, 2 \qquad\qquad \text{for } 1 \leq j \leq J(k)\text{–}1 \tag{36}$$

To develop a similar inequality for $j = J(k)$, first let us justify the following claim:

$$B(x,\, h,\, u_k^{J(k)}) \;\leq\; B(x,\, h,\, v_k^{J(k)}) \;+ 1 \tag{37}$$

If $u_k^{J(k)} = v_k^{J(k)}$, then (37) is obviously true. If $u_k^{J(k)} < v_k^{J(k)}$, then (37) follows from (33). It follows from (37) that

$$P(x, h-1, u_k^{J(k)}, v_k^{J(k)}) \;=\; P(x, h, u_k^{J(k)}, v_k^{J(k)}) + B(x, h, v_k^{J(k)}) - B(x, h, u_k^{J(k)}) \;\geq\; P(x, h, u_k^{J(k)}, v_k^{J(k)}) - 1 \tag{38}$$

Next, the throughput over the entire interval $(s_k\,,\, t_k]$ can be studied. By (30), (36), and (38),

$$P(x,\, h-1,\, s_k\,,\, t_k) \;=\; \left\lceil \sum_{j=1}^{J(k)} P(x,\, h-1,\, u_k^j\,,\, v_k^j) \right\rceil + \left\lceil \sum_{j=1}^{J(k)} P(x,\, h-1,\, v_k^j\,,\, u_k^{j-1}) \right\rceil$$

$$\geq\; \left\lceil \sum_{j=1}^{J(k)} P(x,\, h,\, u_k^j\,,\, v_k^j) \right\rceil + [J(k) - 1]\cdot(W \,-\, 2) \;- 1 \;+ \left\lceil \sum_{j=1}^{J(k)} P(x,\, h-1,\, v_k^j\,,\, u_k^{j-1}) \right\rceil \tag{39}$$

Finally, the throughput over the collection of intervals $(s_1\,,\, t_1]\,,\, \ldots\,,\, (s_K\,,\, t_K]$ can be examined. Summing (39) over k yields:

$$\sum_{k=1}^{K} P(x, h-1, s_k\,, t_k) \geq \left\lceil \sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j\,, v_k^j) \right\rceil + \left\lceil \sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j\,, u_k^{j-1}) \right\rceil + \left\lceil \sum_{k=1}^{K} [J(k)-1] \right\rceil \cdot (W-2) - K \tag{40}$$

The hypotheses of the lemma can now be used to bound the right-hand side of (40). It follows from (30), (34), and assumption (26) that

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x,\, h,\, u_k^j\,,\, v_k^j) \;\geq\; r\cdot \left\lceil \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^j - u_k^j) \right\rceil \;- e'' \;- \left\lceil \sum_{k=1}^{K} J(k) \right\rceil \cdot f'' \tag{41}$$

Similarly, it follows from (30), (31), and assumption (25) that

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x,\, h-1,\, v_k^j\,,\, u_k^{j-1}) \;\geq\; r\cdot \left\lceil \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - v_k^j) \right\rceil \;- e' \;- \left\lceil \sum_{k=1}^{K} J(k) \right\rceil \cdot f' \tag{42}$$

Substituting (41) and (42) into (40) and applying (27) and (30) yields the desired result (28):

$$\sum_{k=1}^{K} P(x,h-1,s_k,t_k) \geq r\cdot\left\lceil \sum_{k=1}^{K}\sum_{j=1}^{J(k)} (u_k^{j-1}-u_k^{j}) \right\rceil - (e'+e'') - K - \left\lceil \sum_{k=1}^{K} J(k) \right\rceil\cdot(f'+f'') + \left\lceil \sum_{k=1}^{K} [J(k)-1] \right\rceil\cdot(W-2)$$

$$= r\cdot\sum_{k=1}^{K}(u_k^0 - u_k^{J(k)}) - (e'+e'') - K\cdot(f'+f''+1) + \left\lceil \sum_{k=1}^{K} [J(k)-1] \right\rceil\cdot(W-f'-f''-2)$$

$$\geq r\cdot\sum_{k=1}^{K}(t_k - s_k) - (e'+e'') - K\cdot(f'+f''+1)$$

□

Proof of Lemma 1

In order to show (3), properties (43) and (44) will be proved.

(43) For each hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for any nondecreasing sequence $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ of times in $[T_1, T_2)$ such that $B(x, h+1, \tau) < W$ for all τ in $\bigcup_{k=1}^{K}[s_k, t_k)$,
$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq r\cdot\sum_{k=1}^{K}(t_k - s_k) - (h+1)\cdot e - K\cdot\left[(h+1)\cdot f + h\right]$

(44) For each hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for any nondecreasing sequence $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ of times in $[T_1, T_2)$ such that $B(x, h, \tau) > 0$ for all τ in $\bigcup_{k=1}^{K}[s_k, t_k)$,
$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq r\cdot\sum_{k=1}^{K}(t_k - s_k) - [H(x)-h+1]\cdot e - K\cdot\left[[H(x)-h+1]\cdot f + H(x) - h\right]$

The proof of (43) is by forward induction on h. The base case (i.e., $h = 0$) follows from assumption (1) and the fact that $P(x, 0, s_k, t_k) = C(x, 0, s_k, t_k)$ during intervals when $B(x, 1, \tau) < W$ (since buffer 0 is never empty). For the induction step, fix a hop h of x in the range $1 \leq h \leq H(x)$. Property (43) is assumed to hold for hop $h-1$, and it will be shown to hold for hop h. Let K be any positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be any times such that

(45) $$T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$$ and

(46) $$B(x, h+1, \tau) < W \text{ for all } \tau \text{ in } \bigcup_{k=1}^{K}[s_k, t_k)$$

The goal is to show that

(47) $$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq r\cdot\sum_{k=1}^{K}(t_k - s_k) - (h+1)\cdot e - K\cdot[(h+1)\cdot f + h]$$

To prove (47), Lemma 2 will be used, with:

$$e' = h \cdot e \qquad e'' = e \qquad f' = h \cdot f + h - 1 \qquad f'' = f$$

First, note that the induction hypothesis can be rephrased in terms of time variables v_k^j and u_k^j as follows:

(48) If $J(1), J(2), \ldots, J(K)$ are any positive integers, and if $v_1^{J(1)}, u_1^{J(1)-1}, v_1^{J(1)-1}, u_1^{J(1)-2}, \ldots, v_1^1, u_1^0, v_2^{J(2)}, u_2^{J(2)-1}, v_2^{J(2)-1}, u_2^{J(2)-2}, \ldots, v_2^1, u_2^0, \ldots, v_K^{J(K)}, u_K^{J(K)-1}, v_K^{J(K)-1}, u_K^{J(K)-2}, \ldots, v_K^1, u_K^0$ is any nondecreasing sequence of times in $[T_1, T_2)$ such that $B(x, h, \tau) < W$ for all τ in $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} [v_k^j, u_k^{j-1})$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^{j-1}) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - v_k^j) - h \cdot e - \left\lceil \sum_{k=1}^{K} J(k) \right\rceil \cdot (h \cdot f + h - 1)$$

Using (48) and (45), it is straightforward to verify condition (25) of Lemma 2. Now condition (26) of Lemma 2 will be verified. As in (26), suppose, for $k = 1, 2, \ldots, K$, that $J(k)$ is some positive integer, and that $u_k^{J(k)}, v_k^{J(k)}, u_k^{J(k)-1}, v_k^{J(k)-1}, \ldots, u_k^1, v_k^1$ is some nondecreasing sequence of times in $[s_k, t_k]$ such that $B(x, h, \tau) > 0$ for all τ in $\bigcup_{j=1}^{J(k)} [u_k^j, v_k^j)$. By (46), then, session x will accept every chance offered to it by the round-robin scheduler at hop h during $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} (u_k^j, v_k^j]$:

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^j) = \sum_{k=1}^{K} \sum_{j=1}^{J(k)} C(x, h, u_k^j, v_k^j)$$

If follows from assumption (1) that

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^j) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^j - u_k^j) - e - \left\lceil \sum_{k=1}^{K} J(k) \right\rceil \cdot f$$

This verifies condition (26). Condition (27) of Lemma 2 is satisfied by assumption (2). All the conditions of Lemma 2 have been verified. Conclusion (29) of Lemma 2 gives the desired result (47).

The proof of (44) is similar to that of (43), but it uses backward induction on h. Details can be found in [41].

Now (3) will be proved. Let h be any hop of x in the range $0 \leq h \leq H(x)$. Let K be any positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be any times satisfying

(49) $$T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$$

If $h = 0$, then (3) follows directly from (44), since $B(x, 0, \tau) > 0$ for all times $\tau \geq 0$; so assume that $h \geq 1$. To

prove (3), Lemma 2 will be used, with:

$$e' = h \cdot e \qquad e'' = [H(x) - h + 1] \cdot e \qquad f' = h \cdot f + h - 1 \qquad f'' = [H(x) - h + 1] \cdot f + H(x) - h$$

First, note that (43) can be applied to hop $h-1$ and rephrased in terms of time variables v_k^j and u_k^j to yield the following property:

(50) If $J(1), J(2), \ldots, J(K)$ are any positive integers, and if $v_1^{J(1)}, u_1^{J(1)-1}, v_1^{J(1)-1}, u_1^{J(1)-2}, \ldots, v_1^1, u_1^0,$ $v_2^{J(2)}, u_2^{J(2)-1}, v_2^{J(2)-1}, u_2^{J(2)-2}, \ldots, v_2^1, u_2^0, \ldots, v_K^{J(K)}, u_K^{J(K)-1}, v_K^{J(K)-1}, u_K^{J(K)-2}, \ldots, v_K^1, u_K^0$ is any nondecreasing sequence of times in $[T_1, T_2)$, and if $B(x, h, \tau) < W$ for all τ in $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} [v_k^j, u_k^{j-1})$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^{j-1}) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - v_k^j) - h \cdot e - \left\lceil \sum_{k=1}^{K} J(k) \right\rceil \cdot (h \cdot f + h - 1)$$

Using (50) and (49), it is straightforward to verify condition (25) of Lemma 2. Next, note that (44) can be rephrased in terms of time variables u_k^j and v_k^j to yield the following property:

(51) If $J(1), J(2), \ldots, J(K)$ are any positive integers, and if $u_1^{J(1)}, v_1^{J(1)}, u_1^{J(1)-1}, v_1^{J(1)-1}, \ldots, u_1^1, v_1^1,$ $u_2^{J(2)}, v_2^{J(2)}, u_2^{J(2)-1}, v_2^{J(2)-1}, \ldots, u_2^1, v_2^1, \ldots, u_K^{J(K)}, v_K^{J(K)}, u_K^{J(K)-1}, v_K^{J(K)-1}, \ldots, u_K^1, v_K^1$ is any nondecreasing sequence of times in $[T_1, T_2)$, and if $B(x, h, \tau) > 0$ for all τ in $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} [u_k^j, v_k^j)$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^j) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^j - u_k^j) - [H(x) - h + 1] \cdot e - \left\lceil \sum_{k=1}^{K} J(k) \right\rceil \cdot \left[[H(x) - h + 1] \cdot f + H(x) - h \right]$$

Using (51) and (49), it is straightforward to verify condition (26) of Lemma 2. Condition (27) of Lemma 2 is satisfied by assumption (2). All the conditions of Lemma 2 have been verified. Conclusion (29) of Lemma 2 gives the desired result (3).

□

REFERENCES

[1] E. L. Hahne and R. G. Gallager, "Round Robin Scheduling for Fair Flow Control in Data Communication Networks," *Proc. IEEE Internatl. Conf. Comm.*, June 1986, pp. 103-107.

[2] M. G. H. Katevenis, "Fast Switching and Fair Control of Congested Flow in Broadband Networks," *IEEE J. Selected Areas Comm.*, Vol. SAC-5, No. 8, Oct. 1987, pp. 1315-1326.

[3] M. Gerla and L. Kleinrock, "Flow Control: A Comparative Survey," *IEEE Trans. Comm.*, Vol. COM-28, No. 4, April 1980, pp. 553-574.

[4] M. Gerla, H. W. Chan, and J. R. Boisson de Marca, "Fairness in Computer Networks," *Proc. IEEE Internatl. Conf. Comm.*, June 1985, pp. 1384-1389.

[5] S. J. Golestaani, "A Unified Theory of Flow Control and Routing in Data Communication Networks," Report LIDS-TH-963, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Jan. 1980.

[6] R. G. Gallager and S. J. Golestaani, "Flow Control and Routing Algorithms for Data Networks," *Proc. Fifth Internatl. Conf. Comp. Comm.*, Oct. 1980, pp. 779-784.

[7] M. Gerla and M. Staskauskas, "Fairness in Flow Controlled Networks," *Proc. IEEE Internatl. Conf. Comm.*, June 1981, pp. 63.2.1-63.2.5.

[8] O. C. Ibe, "Flow Control and Routing in an Integrated Voice and Data Communication Network," Report LIDS-TH-1115, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., August 1981.

[9] H. P. Hayden, "Voice Flow Control in Integrated Packet Networks," Report LIDS-TH-1152, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Oct. 1981.

[10] E. M. Gafni, "The Integration of Routing and Flow-Control for Voice and Data in a Computer Communication Network," Report LIDS-TH-1239, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Sept. 1982.

[11] M. Gerla, H. W. Chan, and J. R. Boisson de Marca, "Routing, Flow Control and Fairness in Computer Networks," *Proc. IEEE Internatl. Conf. Comm.*, May 1984, pp. 1272-1275.

[12] G. H. Thaker and J. B. Cain, "Interactions Between Routing and Flow Control Algorithms," *IEEE Trans. Comm.*, Vol. COM-34, No. 3, March 1986, pp. 269-277.

[13] J. Regnier and P. A. Humblet, "Average Waiting Time Assignment - Part I: The Single Link Case," *IEEE Trans. Comm.*, Vol. COM-38, No. 11, Nov. 1990, pp. 2049-2059.

[14] J. Regnier and P. A. Humblet, "Average Waiting Time Assignment - Part II: The Integrated Services Network Case," *IEEE Trans. Comm.*, Vol. COM-38, No. 11, Nov. 1990, pp. 2060-2072.

[15] J. P. Sauve, J. W. Wong, and J. A. Field, "On Throughput and Fairness in Packet Switching Networks with Window Flow Control," CCNG Report E-100, Computer Communications Networks Group, Univ. of Waterloo, Waterloo, Ontario, Canada, Dec. 1981.

[16] J. P. Sauve, J. W. Wong, and J. A. Field, "Improving Total Throughput in Packet Switching Networks with Window Flow Control," *Proc. IEEE Global Telecomm. Conf.*, Nov.-Dec. 1982, pp. 1189-1194.

[17] K. Bharath-Kumar and J. M. Jaffe, "A New Approach to Performance-Oriented Flow Control," *IEEE Trans. Comm.*, Vol. COM-29, No. 4, April 1981, pp. 427-435.

[18] J. M. Jaffe, "Flow Control Power is Nondecentralizable," *IEEE Trans. Comm.*, Vol. COM-29, No. 9, Sept. 1981, pp. 1301-1306.

[19] T. Bially, B. Gold, and S. Seneff, "A Technique for Adaptive Voice Flow Control in Integrated Packet Networks," *IEEE Trans. Comm.*, Vol. COM-28, No. 3, March 1980, pp. 325-333.

[20] J. M. Jaffe, "A Decentralized, "Optimal", Multiple-User, Flow Control Algorithm," *Proc. Fifth Internatl. Conf. Comp. Comm.*, Oct. 1980, pp. 839-844.

[21] J. M. Jaffe, "Bottleneck Flow Control," *IEEE Trans. Comm.*, Vol. COM-29, No. 7, July 1981, pp. 954-962.

[22] E. M. Gafni and D. P. Bertsekas, "Dynamic Control of Session Input Rates in Communication Networks," *IEEE Trans. Auto. Control*, Vol. AC-29, No. 11, Nov. 1984, pp. 1009-1016.

[23] D. A. Oshinsky, "Use of Fair Rate Assignment Algorithms in Networks with Bursty Sessions," S. M. Thesis, Dept. of Elec. Engr. and Comp. Sci., Mass. Inst. of Technology, Cambridge, Mass., May 1984.

[24] J. Mosely, "Asynchronous Distributed Flow Control Algorithms," Report LIDS-TH-1415, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Oct. 1984.

[25] H. Luss and D. R. Smith, "Resource Allocation among Competing Activities: A Lexicographic Minimax Approach," *Oper. Res. Letters*, Vol. 5, No. 5, Nov. 1986, pp. 227-231.

[26] U. Mukherji, "A Schedule-Based Approach for Flow-Control in Data Communication Networks," Report LIDS-TH-1527, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Jan. 1986.

[27] U. Mukherji, "A Schedule-Based Approach for Flow-Control in Data Communication Networks," *Proc. IEEE Global Telecomm. Conf.*, Dec. 1986, pp. 98-104.

[28] A. G. Fraser, "Towards a Universal Data Transport System," *IEEE J. Selected Areas in Commun.*, Vol. SAC-1, No. 5, November 1983, pp. 803-816.

[29] S. P. Morgan, "Window Flow Control on a Trunked Byte-Stream Virtual Circuit," *IEEE Trans. Comm.*, Vol. COM-36, No. 7, July 1988, pp. 816-825.

[30] S. P. Morgan, "Queueing Disciplines and Passive Congestion Control in Byte-Stream Networks," *Proc. IEEE INFOCOM '89*, April 1989, pp. 711-720, also to appear in *IEEE Trans. Comm.*

[31] S. S. Panwar, T. K. Philips, and M.-S. Chen, "Golden Ratio Scheduling for Low Delay Flow Control in Computer Networks," *Proc. IEEE Global Telecomm. Conf.*, Dec. 1988, pp. 1113-1118.

[32] C. R. Kalmanek, H. Kanakia, and S. Keshav, "Rate Controlled Servers for Very High-Speed Networks," *Proc. IEEE Global Telecomm. Conf.*, December 1990.

[33] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm," *Proc. ACM SIGCOMM '89*, Sept. 1989, pp. 1-12.

[34] A. Greenberg and N. Madras, "Comparison of a Fair Queueing Discipline to Processor Sharing," *Proc. Performance '90*, Sept. 1990, pp. 193-207.

[35] J. R. Davin and A. T. Heybey, "A Simulation Study of Fair Queueing and Policy Enforcement," *Comput. Commun. Rev.*, Vol. 20, No. 5, Oct. 1990, pp. 23-29.

[36] L. Zhang, "A New Architecture for Packet Switching Network Protocols," Ph.D. Thesis, Dept. of Elec. Engr. and Comp. Sci., Mass. Inst. of Technology, Cambridge, Mass., July 1989.

[37] E. L. Hahne, "Round-Robin Scheduling and Window Flow Control for Max-Min Fairness in Data Networks," technical report, November 1987.

[38] F. Baskett et al., "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *J. ACM*, Vol. 22, No. 2, April 1975, pp. 248-260.

[39] A. J. Lemoine, "Networks of Queues — A Survey of Equilibrium Analysis," *Mgt. Sci.*, Vol. 24, No. 4, Dec. 1977, pp. 464-481.

[40] M. Reiser, "A Queueing Network Analysis of Computer Communication Networks with Window Flow Control," *IEEE Trans. Comm.*, Vol. COM-27, No. 8, Aug. 1979, pp. 1199-1209.

[41] E. L. Hahne, "Round Robin Scheduling for Fair Flow Control in Data Communication Networks," Report LIDS-TH-1631, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Dec. 1986.

[42] R. L. Cruz, "Maximum Delay in Buffered Multistage Interconnection Networks," *Proc. IEEE INFOCOM '88*, March 1988, pp. 135-144.

[43] P. Billingsley, *Probability and Measure*, New York: Wiley, 1986.

[44] M. Iosifescu, *Finite Markov Processes and Their Applications*, New York: Wiley, 1980.

[45] J. D. C. Little, "A Proof of the Queueing Formula $L=\lambda W$," *Operations Research*, Vol. 9, No. 3, May-June 1961, pp. 383-387.

[46] T. P. Barzilai et al., "Adaptive Session-Level Pacing," U. S. Patent No. 4736369, April 5, 1988.

[47] E. L. Hahne, C. R. Kalmanek, and S. P. Morgan, "Fairness and Congestion Control on a Large ATM Data Network," *Proc. 13th International Teletraffic Congress*, June 1991.

[48] V. Jacobson, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM '88*, Aug. 1988, pp. 314-329.

[49] B. T. Doshi and H. Heffes, "Overload Performance of an Adaptive, Buffer-Window Allocation Scheme, with Reservation Renewal, for a Class of High Speed Networks," *Proc. 13th International Teletraffic Congress*, June 1991.

[50] D. Mitra and J. B. Seery, "Dynamic Adaptive Windows for High Speed Data Networks: Theory and Simulations," *Proc. ACM SIGCOMM '90*, Sept. 1990, pp. 30-40.

[51] D. Mitra and J. B. Seery, "Dynamic Adaptive Windows for High Speed Data Networks with Multiple Paths and Propagation Delays (extended abstract)," *Proc. IEEE INFOCOM '91*, April 1991, pp. 39-49.

[52] R. Jain, "A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks," *Comput. Commun. Rev.*, Vol. 19, No. 5, Oct. 1989, pp. 56-71.

[53] R. Jain, "A Timeout-Based Congestion Control Scheme for Window Flow-Controlled Networks," *IEEE J. Selected Areas in Commun.*, Vol. SAC-4, No. 7, Oct. 1986, pp. 1162-1167.

[54] K. K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with a Connectionless Network Layer," *Proc. ACM SIGCOMM '88*, Aug. 1988, pp. 303-313.

[55] D. T. D. Luan and D. M. Lucantoni, "Throughput Analysis of a Window-Based Flow Control Subject to Bandwidth Management," *Proc. IEEE INFOCOM '88*, March 1988, pp. 411-417.

[56] S. Keshav, A. K. Agrawala, and S. Singh, "Design and Analysis of a Flow Control Algorithm for a Network of Rate Allocating Servers," *Proc. IFIP WG 6.1/6.2 2nd Internatl. Wkshp. on Protocols for High Speed Netwks.*, Nov. 1990.

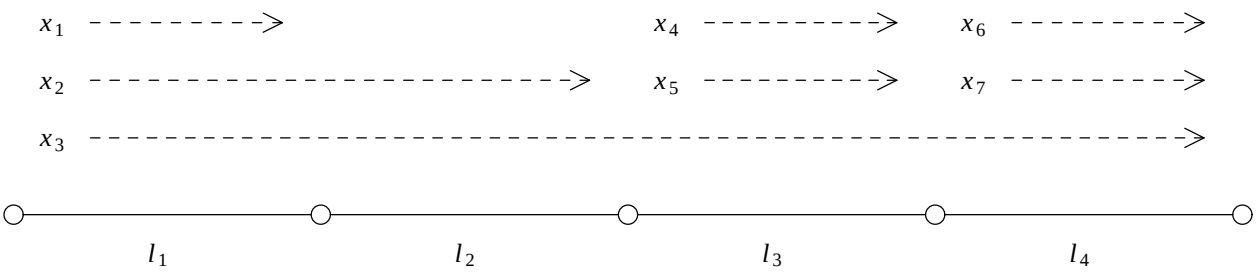

Figure 1. Defining Max-Min Fairness.

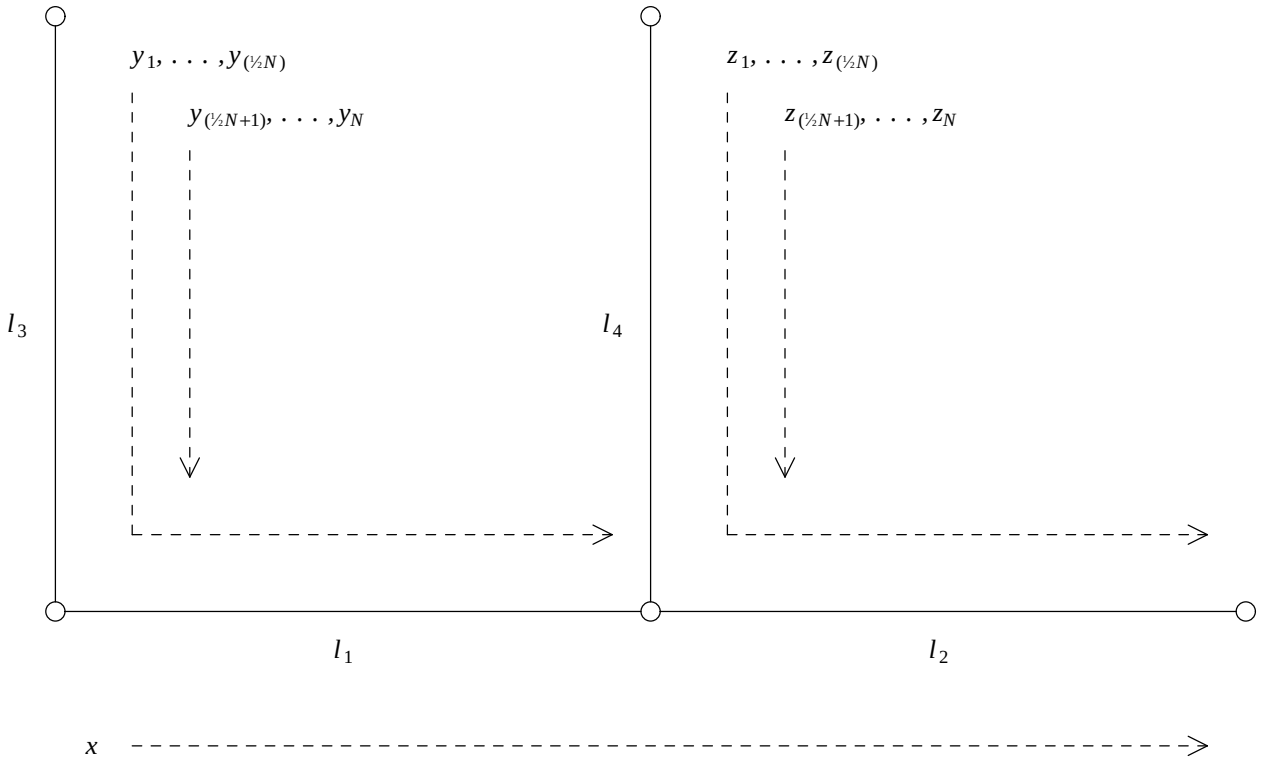

Figure 2. An Example of Unfairness.

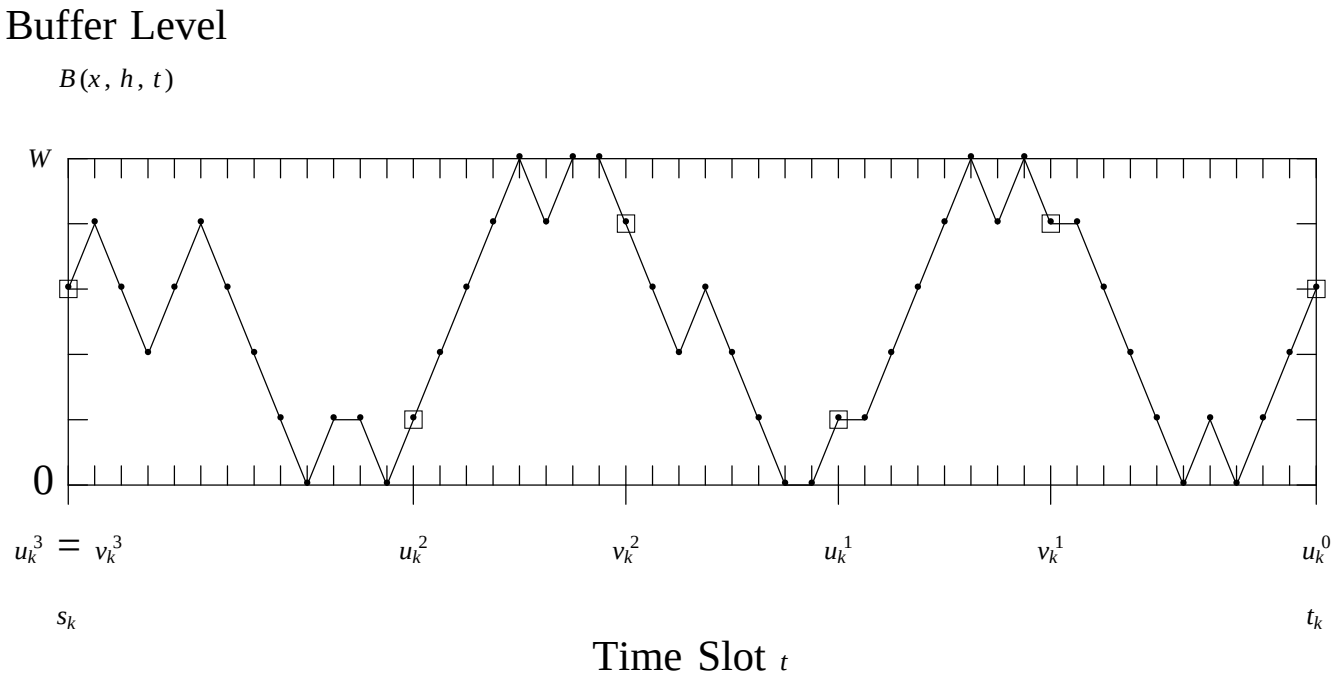

Figure 3. Locating u_k^j and v_k^j .