

Scheduling to Minimize the Worst-Case Loss Rate

Mahmoud Elhaddad[†] Hammad Iqbal[‡] Taieb Znati^{†,‡} Rami Melhem[†]

[†]Department of Computer Science

[‡] Department of Information Sciences and Telecommunications
University of Pittsburgh

Abstract

We study link scheduling in networks with small router buffers with the goal of minimizing the guaranteed packet loss rate bound for each ingress–egress traffic aggregate (connection). Given a link scheduling algorithm (a service discipline and a packet drop policy), the guaranteed loss rate for a connection is the loss rate under worst-case routing and bandwidth allocations for competing traffic. We show that a *local* min-max fairness property with respect to apportioning loss events among the connections sharing each link, and the correlation of scheduling decisions at different links are two necessary and (together) sufficient conditions for optimality in the minimization problem. Based on these conditions, we introduce and analyze a randomized link-scheduling algorithm called Rolling Priorities (RP) where packet scheduling at each link relies exclusively on local information. We show that RP satisfies both conditions and is therefore optimal. Furthermore, we show that the algorithm combining FCFS with the Random Drop policy (FCFS/RD) is locally fair and that it is nearly optimal under light link load. Under heavy load, the guaranteed loss rate under FCFS/RD deteriorates much faster as a function of path length compared to the optimal algorithm.

Our study is motivated by the challenge of providing statistical loss rate guarantees to traffic aggregates traversing networks with small buffers, without sacrificing network utilization. Given a desired bound on the loss rate of every connection, each scheduling algorithm imposes constraints on the maximum link utilization and the maximum routing path length. We compare the performance of RP, FCFS/RD, FCFS/DropTail and Round-Robin scheduling using simulation. Results indicate that the optimal algorithm, RP, results in significantly less restrictions on connection routing.

1 Introduction

1.1 Motivation and Problem

Due to increasing data rates, and the drive towards constructing photonic packet switches with integrated optical packet buffers, in the near future Internet routers are expected to have limited buffering capacity [24, 4, 14, 5]. Unfortunately, when the buffer size at a link is limited to dozens of packets, the packet loss rate at that link can be as high as 10^{-2} under light load. Recent research [14] has shown that loss-sensitive TCP flows traversing a single work-conserving link having a small buffer are able to withstand high loss rate and achieve good link utilization, under assumptions that limit the contribution of each flow to the total link

load.¹ However, several questions regarding the performance of networks with small router buffers remain open [11]. This research is motivated by one question that is critical to the usability and dependability of such networks:

What statistical guarantees on the packet loss rate for users (flows or aggregates thereof) can be supported by a network having small router buffers without severely restricting the maximum allowable link utilization or the maximum path length?

Loss guarantees are essentially statistical bounds on the loss rate experienced by a flow or an aggregate of flows. The bounds depend on the path length and the load at the links it traverses, and are obtained assuming worst-case traffic scenarios. The study of such guarantees is of value to Internet Service Providers (ISPs) offering guaranteed-bandwidth services with statistically bounded packet loss rate. Such service guarantees are commonly offered since ISPs cannot require their clients not to use loss-sensitive TCP or its variants for high-speed data transfers, or require them to employ only loss-tolerant applications (e.g., through Forward Error Correction). Since the loss rate of a flow is a function of the path length and the load at the links the flow traverses, providing loss guarantees imposes path length and link utilization constraints on the paths that the network may use to route traffic. Obviously, these constraints limit the traffic-carrying capacity of the network. Nevertheless, accepting traffic based on routing constraints derived from worst-case bounds on the loss rate is necessary, if the network is to provide loss guarantees.

Given the load at the network links and the link buffer capacities, the loss rate along a network path is determined by three factors: (1) the packet arrival process, (2) the packet size distribution, and (3) the scheduling algorithm (service discipline and packet drop policy) used at the links. The effect of variability in the arrival process and the benefit of limiting burstiness by regulating the arrival process have been well studied and understood [27]. Similar results are known for the distribution of packet sizes [19]; constant packet sizes are desirable when the objective is to minimize the packet loss rate at a link. In contrast, there are only few known results concerning the performance of scheduling algorithms in networks with fixed-size buffers, where the objective is to minimize the loss rate [2].

Motivated by the above question, we study the problem of link scheduling to minimize the guaranteed on packet loss rate bound for each ingress–egress traffic aggregate (connection), given its path length and the load at the links. For practical relevance, we restrict our investigation to algorithms that are work-conserving², and *local* in the sense that scheduling decisions are based only on local information. The FCFS/DT algorithm combining the First-Come-First-Served discipline and the Drop-Tail policy is the most common example of local work-conserving algorithms.

A scheduling problem instance on a given network is defined by a set of connections (ingress–egress traffic aggregates), each assigned a fixed path and a fixed bandwidth allocation such that the load at the links does not exceed a parameter $\rho \leq 1$.³ Connections inject packets into the network according to a stochastic joint arrival process. Given a link scheduling algorithm, the guaranteed loss rate for a connection is the

¹The packet loss rate at a link is the number of lost packets as a fraction of the total number averaged over some interval of time. The packet loss rate of a flow or flow aggregate over the path it traverses is defined in a similar way.

²A work conserving algorithm is one that never leaves the link idle while there are packets in the buffer, and never drops packets when there is room in the buffer.

³The load at a link is the ratio of the total bandwidth allocation for connections traversing the link to the link’s capacity.

loss rate under worst-case routing and bandwidth allocations for competing traffic along its path, without violation of the link load constraint, ρ .

An algorithm that minimizes the guaranteed loss rate given the path length and the maximum load at the links (the objective in this paper) is one that imposes the least restrictions on connection routing (in terms of load and path length) to achieve a desired loss guarantee. Specifically, there are problem instances where a link scheduling algorithm that seeks to further minimize the maximum loss rate by giving service priority to connections traversing a larger number of hops does not yield better bounds for “long-haul” connections: the cases where connections sharing each link have (nearly) equal path lengths. Excluding such cases would complicate the routing problem by requiring additional constraints on the composition of traffic traversing every link.

The networks under consideration have time-slotted links with fixed slot size (in bits) and possibly different link capacities. Incoming packets at the ingress routers are classified into ingress–egress traffic aggregates (connections) and, given the loss rate minimization objective, they are packed into time-slot sized packets before being injected into the network. Links are output buffered, as is commonly assumed in literature on scheduling in packet-routing networks (related work is reviewed in §1.3). It should be noted that results for networks with Output Queued routers carry over to networks with Combined Input-Output Queuing via results in [9] and [5].

1.2 Results and Contributions

We show that a *local* min-max fairness property with respect to apportioning loss events among the connections sharing each link, and a condition on the correlation of scheduling decisions at different links are necessary and together sufficient for optimality in the minimization problem. Algorithms that are locally min-max fair are referred to simply as *locally-fair*. The correlation property refers to packets from the same connection having consistent “priorities” at every hop in such a way that the maximum possible fraction of packets experience low loss rate throughout the path.

Based on the optimality conditions, we introduce and analyze a randomized link-scheduling algorithm called Rolling Priorities (RP) where packet scheduling at each link relies exclusively on local information. We show that RP satisfies both conditions and is therefore optimal. Furthermore, we show that the algorithm combining FCFS with the Random Drop policy (FCFS/RD) is locally fair and that it is nearly optimal under light link load. Under heavy load, the guaranteed loss rate under FCFS/RD deteriorates much faster as a function of path length compared to the optimal algorithm.

With high probability, RP ensures that the largest possible fraction of packets from each connection is subject to a small loss rate at every link, irrespective of the average link loss rate. From the perspective of a connection, time is divided into epochs of fixed duration. The connection’s scheduling priority at every link is low at the beginning of an epoch and improves as time progresses, until the start of a new epoch. The key to the high probability argument is that random choice of the “phase” of each connection, done once during connection initialization, makes it unlikely that a large number of connections sharing a link have synchronized epochs (i.e., simultaneously have high priority), provided that the duration of an epoch is sufficiently large in relation to the number of connections sharing each link.

We provide numerical and simulation examples comparing the performance of RP and FCFS/RD to FCFS/DT and round-robin (fair queuing) scheduling. Results confirm that scheduling algorithms having the local-fairness property result in significantly less restrictions on connection routing under light and moderate load.

1.3 Related Work

Until recently, the performance of scheduling algorithms in packet networks has mostly been investigated in terms of packet delay and stability (i.e., boundedness of backlog) guarantees. These studies, for example [20, 7, 28, 22, 10, 19, 21, 6], have led to valuable insights into the behavior of service disciplines such as FCFS and Processor Sharing [20, 25, 3]. However, in investigating delay and stability, the packet network is modeled as a queuing network where communication links are represented by servers with infinite waiting room, which limits the practical value of the resulting algorithmic guarantees when applied to network with limited buffering capacity.

Although delay and stability guarantees lead to bounds on buffer occupancy that can be leveraged in dimensioning buffer capacities at the links to prevent (or at least bound) packet loss, the occupancy bounds are often dependent on network parameters, such as the diameter of the network and link capacities, which are impractical to track in today’s large decentralized networks. More importantly, by relying on such bounds for buffer dimensioning, one would be ignoring the technological constraints on buffer capacity which recently arose due to increasing link speeds [4], and the drive towards constructing photonic packet switches with integrated optical packet buffers [24, 8, 5].

The work by Reisslein et al. [26] provides a bufferless-multiplexing framework for supporting statistical delay guarantees in multihop networks. Using traffic regulation at the ingress and bufferless multiplexing at the core, they transform the problem of providing ingress–egress delay guarantees into one of providing loss guarantees. The loss bounds are obtained using an approximate fluid-multiplexer model. The fluid model may severely underestimate the loss probability in packet multiplexers (links) because of the assumption that flows can have a fixed peak transmission rate (smaller than the link capacity) across all time scales. Under this assumption, a bufferless fluid multiplexer can simultaneously serve multiple flows without incurring “fluid” loss. Note that on the other hand, a packet multiplexer (a link) can only serve one packet (thus one flow) at a time at time scales smaller than the packet transmission time. The scheduling order (service discipline) is trivial in the bufferless multiplexing model. For the drop policy, the authors assume that if at any instant the sum of flow rates exceeds the capacity of the link, fluid loss is shared proportionally among flows. As we shall see, this is not generally satisfied by packet drop policies.

Under buffer capacity constraints, the packet loss rate is the primary metric in the evaluation of scheduling algorithms. Surprisingly, to date this area of research has remained largely unexplored, with only few known results. Results are known only for the problem of maximizing the overall network throughput (maximizing the number of successfully delivered packets within a given time interval) under adversarial packet injection [2, 18]), but not for user-oriented throughput or loss metrics.

Active Queue Management (AQM) schemes [15], most notably Random Early Detection (of congestion) (RED) [16] attempt to prevent loss synchronization and fairly apportion loss among TCP flows sharing

a common bottleneck by voluntarily (probabilistically) dropping packets without buffer overflow. RED has been evaluated on a single bottleneck with small buffer and was shown to perform poorly in this setting [23]. The reason however is shared among all AQM schemes which are designed to detect the onset of congestion (overload) by observing the buffer occupancy using a moving average over a long time interval, rather than observing the instantaneous queue length. These schemes are too slow to react when the buffer capacity is small such that loss occurs without persistent overload (Here we assume links are not overloaded). Algorithms that apportion loss fairly at every link are represented in this paper by FCFS/RD.

To the best of our knowledge, this work is the first to investigate local scheduling algorithms for providing *per-session* loss guarantees in packet networks with bounded buffers. For networks using advance transmission reservations, we previously presented and a reservation scheduling algorithm and quantified its blocking guarantees assuming a particular arrival process for the reservation requests [13, 12].

1.4 Paper Organization

The paper is organized as follows: In the next section we present the formulation of the loss rate minimization problem and state our assumption regarding the traffic arrival process. In Section 3, we define local fairness and show that it is a necessary condition for optimality in the minimization problem. We also discuss why it is not readily satisfied by every link scheduling algorithm. Then, we identify another necessary condition on the statistical correlation of scheduling decisions, and establish that together the conditions are sufficient for optimality. In Section 4, we present the Rolling Priority algorithm and discuss its implementation issues. Analysis of Rolling Priority is presented in Section 5, where we establish its optimality. In Section 6, we analytically relate the performance of FCFS/RD to that of the optimal algorithm. This is followed by numerical and simulation results in Sections 7 and 8, and concluding remarks in Section 9.

2 Preliminaries and Problem Statement

In this section we introduce notation and assumptions leading to a formal statement of the problem.

2.1 The loss rate of a connection and the aggregate loss rate at a link

Consider a link l shared by N connections, labeled 1 through N and suppose packets arrive at l according to a known stochastic arrival process jointly defined for all connections. Let $A_i^l(t_1, t_2)$ be a random variable representing the number of packet arrivals due to a connection i during the interval $[t_1, t_2)$, and let $X_i^{l,G}(t_1, t_2) \leq A_i^l(t_1, t_2)$ be the number of packet losses at l among the $A_i^l(t_1, t_2)$ arrivals if the link uses scheduling algorithm G . The loss rate for connection i over the same interval is given by

$$R_i^{l,G}(t_1, t_2) \triangleq \frac{X_i^{l,G}(t_1, t_2)}{A_i^l(t_1, t_2)}.$$

For a given sample path s of the joint packet arrival process (packet inject sequence), the number of packets lost from connection i is denoted $X_i^{s,l,G}(t_1, t_2)$. The value of $X_i^{s,l,G}(t_1, t_2)$ is deterministic if G employs

deterministic service and drop policies. Otherwise it is a random variable that reflects the random choices of the algorithm. The loss rate under sample path s is similarly denoted $R_i^{s,l,G}$ and is by definition a random variable (rv) whenever $X_i^{s,l,G}(t_1, t_2)$ is an rv.

When the buffer capacity at the links is small, the above notation can be extended to connections routed over multihop paths based on the following assumption:

Assumption 1. *Consider an arbitrary connection c routed along a path π and let Δ_c^l be the propagation delay along π up to link l . We assume that for each sufficiently large interval $[t_1, t_2)$ and every work-conserving scheduling algorithm, the number of connection c 's packets injected at the ingress in $[t_1, t_2)$ that arrive at any $l \in \pi$ after $t_2 + \Delta_c^l$ is negligible.*

For the assumption to hold, the length of the interval must be large compared to the buffer size so that link busy periods are unlikely to extend beyond the end of the interval. The assumption rules out adversarial sources which, for any decomposition of the time axis into contiguous intervals, may choose to inject packets only at the end of some or all intervals.

We denote the number of arrivals at link l in the interval $[t_1 + \Delta_c^l, t_2 + \Delta_c^l)$ by $\tilde{A}_c^{l,G}(t_1, t_2)$. We apply the same convention to denote the number of lost packets $\tilde{X}_c^{l,G}(t_1, t_2)$ and the loss rate at the link $\tilde{R}_c^{l,G}(t_1, t_2)$.

The path loss rate experienced by the connection among packets injected at the ingress during $[t_1, t_2)$ is defined as:

$$\begin{aligned} R_c^G(t_1, t_2) &\triangleq \frac{1}{A_c(t_1, t_2)} \sum_{l \in \pi} \tilde{X}_c^{l,G}(t_1, t_2) \\ &= \frac{1}{A_c(t_1, t_2)} \sum_{l \in \pi} \tilde{A}_c^{l,G}(t_1, t_2) \tilde{R}_c^{l,G}(t_1, t_2) \end{aligned} \quad (1)$$

where $A_c(t_1, t_2)$ denotes the number of packets offered by connection c in $[t_1, t_2)$. As with the single-link case, for a given sample path of the joint arrival process of all connection sharing links with c , the loss rate of connection c over the packet injection interval $(t_1, t_2]$, $R_c^{s,l,G}(t_1, t_2)$ is not an random variable unless the scheduling algorithm employed at the links is randomized. The following are two basic results that are used later in the paper:

Monotonic dependence of connection loss rate on the loss rate at the links

Suppose we fix an arrival sample path and exchange the scheduling algorithm at every hop along the path of a connection with one that better favors it in service and drop decisions, thus improving the loss rate at every link. Naturally, the overall (path) loss rate of the connection would improve. For example, suppose a connection, c , that has the shortest path among the connections sharing links in its path, and that all links originally use the Furthest-To-Go protocol which favors packets from connections traversing long paths. Replacing the Furthest-To-Go algorithm with Nearest-To-Go would improve the overall loss rate for connection c . This intuitive result can be easily established by induction along the path of the connection and observing that the number of total packets lost up to any link along the path is higher under the algorithm of higher loss rate at every link. Since the result applies to any sample path, it also applies in expectation. This property is stated in the following lemma whose proof is omitted:

Lemma 1. For any connection c and scheduling algorithms G, G' if $\mathbf{E}[R_c^{l,G}] \leq \mathbf{E}[R_c^{l,G'}]$ at every link l along c 's path. Then $\mathbf{E}[R_c^G] \leq \mathbf{E}[R_c^{G'}]$.

Local equivalence of work-conserving algorithms

Define a busy period $[t_1, t_2]$ at a link l as an interval of slots such that at the end of slot $t_1 - 1$ no packets are available in the buffer, more than one packet arrives at the beginning of t_1 , and the buffer is not empty (excluding the packet being serviced by the link) at the end of every $t \in [t_1, t_2]$, except t_2 . Let $[t_1, t_2]$ be a busy period of link l . Then given the sequence of arrivals during the busy period, the number of packets dropped in the busy period (hence the loss rate) is constant for all work-conserving algorithms whether deterministic or randomized. Furthermore, since a work-conserving algorithm does not drop packets when there is buffer space available, the buffer occupancy at any $t \in [t_1, t_2]$ is the same under all work-conserving algorithms. Starting with the first busy period of the link inductively yields the following lemma:

Lemma 2. Consider a link l subject to a particular sample path of the joint arrival process of the connections traversing it. Then for any interval $[t_1, t_2)$ the loss rate at the link is the same under all work-conserving algorithms.

Let the aggregate loss rate at link l under algorithm G be denoted by $R^{l,G}$ and suppose the link is shared among N connections labeled 1 through N . Then

$$R^{l,G}(t_1, t_2) = \frac{1}{\sum_i A_i^{l,G}(t_1, t_2)} \sum_{i=1}^N A_i^{l,G}(t_1, t_2) R_i^{l,G}(t_1, t_2).$$

Since Lemma 2 holds for every arrival sample path, we can write:

$$R^{l,G}(t_1, t_2) = R^{l,G'}(t_1, t_2)$$

for every pair of work-conserving algorithms G and G' and every interval $[t_1, t_2)$.

Now we formally state the loss rate minimization problem.

2.2 The minimization problem

For a given algorithm G , a fixed arrival sample path s , and an interval $I = [t_1, t_2)$, define $M_c^{s,G}(I)$ as the loss rate of connection c when, at every link l along the path of the connection, $R_c^{s,l,G}(I)$ is the maximum loss rate at l among all connections sharing the link. Using an argument similar to that of Lemma ??, $M_c^{s,G}(I)$ is at least as large as $R_c^{s,G}(I)$. It is a tight bound in the sense that $M_c^{s,G}(I) = R_c^{s,G}(I)$ when G is an algorithm using service and drop policies based on implicit or explicit priority that always favor connection c less than the competing connections. The most obvious example is scheduling based on preassigned connection priorities, where all packets from a given connection have the same fixed priority. FCFS/DT (drop tail) can be considered an implicit priority algorithm when traffic arrivals at a link are partially synchronized such that packets from one or more connection tend to arrive to a full buffer. We will revisit this issue of scheduling bias under FCFS/DT and how common it is in the next section.

In general form, the loss rate minimization problem is stated as follows. Find an algorithm G such that: For every network connection c and interval I , $\mathbf{E}[M_c^G(I)]$ is minimum. The expectation is defined over the distribution of joint arrival process and the choices of algorithm G if it is randomized.

We limit our attention to the case where, for individual connections, the arrival process at every link is a generalization of the Poisson process. Specifically, we assume that, at every link, the packet arrival process due to every connection satisfies the following assumption:

Assumption 2. *There exist $T_0 > 0$ such that for each connection (a) the number of packet arrivals at any link follows an identical probability distribution over disjoint intervals of length T_0 , and (b) the arrivals within the disjoint intervals are not negatively correlated.*

Note that this assumption allows a connection to exhibit a different arrival process at different links hence does not exclude changes in the process due to buffering at intermediate links. One can immediately see that Poisson packet arrivals satisfy this assumption for every T_0 . This model also holds for connections multiplexing periodic (CBR-like) traffic sources such as voice flows, where T_0 can be the time between consecutive packets from a single flow. More importantly, it applies to connections multiplexing bursty flows (e.g., TCP) where burst arrival can be modeled as a Poisson process (See for example [14]). T_0 in this case is the largest burst interarrival time among all connections. The condition on negative correlation is needed to reveal the role played by the length of the optimization interval. Specifically, it excludes traffic processes that inject packets in a purely adversarial manner that prevents optimization.

Let $\mathcal{I}_n, n \geq 1$ denote the set of time intervals of length nT_0 , and let \mathcal{C} the set of connections in the network. We define the set of (work-conserving local-control) algorithms, G , that minimize $\mathbf{E}[M_c^G]$ over all intervals in \mathcal{I}_n for every connection as:

$$\text{MML}(n) = \{G : G \text{ minimizes } \mathbf{E}[M_c^G(I)] \quad \forall I \in \mathcal{I}_n, \forall c \in \mathcal{C}\}$$

where the expectation is defined over the random choices of the algorithm at different links and the distribution of the joint arrival process. We refer to algorithms in $\text{MML}(n)$ for a given n as optimal algorithms for the minimization problem.

In the next section, we seek necessary and sufficient conditions for optimality under the following simplifying assumption:

Assumption 3. *Let G and G' be any two work-conserving algorithms. Then for any arrival sample path, exchanging G and G' at any network link does not increase the aggregate loss rate at any network link.*

This assumption is justifiable for work-conserving algorithms in a network with small buffers. When buffers are large, the change of scheduling algorithms may introduce burstiness that significantly increases the loss rate at downstream links.

3 Optimality Conditions

3.1 Local fairness

Here, we relate the minimization problem above to a problem of fairly apportioning losses among connections sharing each link. We find that fair apportioning of losses at every link is necessary for minimizing the expected loss rate for all connections in a network with small buffers.

Although Lemma 2 implies that at a given link l the expected aggregate loss rate $\mathbf{E}[R^{l,G}(I)]$ is the same under all work-conserving algorithms, connections may have different expected loss rates at the link. For example, under the Nearest-To-Go scheduling algorithms, connections with fewest number of remaining hops to traverse will have smaller loss rate at the expense of other connections.

We define the set of “locally-fair” algorithms over intervals of length nT_0 , $\text{LF}(n)$ as follows:

Definition 1. (Local fairness) *Consider a link l shared by N connections $1, 2, \dots, N$ and employing a work-conserving scheduling algorithm G . $G \in \text{LF}(n)$ if for all $I \in \mathcal{I}_n$ G minimizes the maximum expected loss rate among all connections. That is:*

$$\text{LF}(n) = \{G : G \text{ minimizes } \max_{1 \leq i \leq N} \mathbf{E}[R_i^{l,G}(I)]\},$$

where the expectation is defined over the joint arrival distribution and the decisions of the algorithm at the different link.

It is easy to see that if $G \in \text{LF}(n)$ then $G \in \text{LF}(n')$ where n' is an integral multiple of n . In particular, if $G \in \text{LF}(1)$ then $G \in \text{LF}(n)$ for every $n \geq 1$.

The Nearest-To-Go algorithm is clearly not in $\text{LF}(n)$ for any n . Perhaps more subtly, neither is the FCFS/DT (drop-tail) algorithm. FCFS/DT suffers from a phenomenon referred to as the “traffic-phase effects” where some connections may persistently experience much higher loss rate than other connections sharing the link. This phenomenon, in some instances called the buffer-lockout problem, is due to synchronization of traffic from different connections, which occurs when the link is multiplexing regulated traffic or TCP flows [17]. Thus FCFS/DT is generally not in $\text{LF}(n)$ for any n . However, $\text{FCFS/DT} \in \text{LF}(1)$ under Poisson packet arrivals. This is due to the PASTA property which entails that the expectation of the loss rate is identical for all connections, equal to the expectation of the aggregate loss rate at the link.⁴

We now show that any optimal scheduling algorithm for the minimization problem must be locally fair.

Theorem 1. *Suppose Assumption 3 holds. Then for every $n \geq 1$, $\text{MML}(n) \subseteq \text{LF}(n)$.*

Proof. For any $n \geq 1$, consider a connection c routed over a path π and suppose the network uses a scheduling algorithm $G' \notin \text{LF}(n)$ at every link. If G' is replaced by $G \in \text{LF}(n)$ at every $l \in \pi$. Since G has to be work-conserving, Assumption 3 implies that the expected loss rate at every l over any interval $I \in \mathcal{I}_n$

⁴Possible correlation of traffic arrivals among subsets of connections prevents the definition of local fairness as the case where all connections have the same expected loss rate. Clearly, connections with synchronized arrivals would have higher expected loss rate under any work-conserving algorithm compared to a connection whose arrivals are negatively correlated or independent of arrival from other connections.

remains unchanged. Furthermore, since $G \in \text{LF}(n)$, at every l , the difference of the maximum expected loss rate among connections sharing l from the expected loss rate at l either decreases or remain unchanged. Consequently, the maximum expected loss rate under G is smaller at every l compared to G' . By Lemma 1, we get $\mathbf{E}[M_c^G(I)] \leq \mathbf{E}[M_c^{G'}(I)]$. □

Two questions that naturally arise are: (1) Whether there are work-conserving algorithms that are locally fair for arrival processes satisfying Assumption 2 (in $\text{LF}(n)$, for some $n \geq 1$), and (2) whether local fairness is a sufficient condition for optimality in the minimization problem. In the following subsection we show that FCFS/RD (random drop) is in $\text{LF}(n)$ for any $n \geq 1$. Then we show that local fairness is not a sufficient condition for optimality by identifying two additional necessary conditions that are not satisfied by FCFS/RD. Together with local fairness, these conditions are shown to be sufficient for optimality.

FCFS/RD is locally fair

Under FCFS/RD, the Random Drop policy is as follows: Suppose the buffer size is $B \geq 0$. If a packet arrives at time slot τ to a full buffer (just a busy link in case $B = 0$), a “victim” packet is chosen at random from the set of packet available at the link but not in service, including the arriving packet. This packet is then dropped. That is, each packet—including the new arrival—is dropped with probability $1/(B + 1)$.

Theorem 2. *FCFS/RD $\in \text{LF}(n)$ for every $n \geq 1$.*

Proof. It suffices to show that $\text{FCFS/RD} \in \text{LF}(1)$. That is, it minimizes the maximum expected loss rate among connections sharing a link under every sample path of the joint arrival process and over every interval of length T_0 . The lemma immediately follows by unconditioning on the arrival sample path.

Consider an interval of length T_0 on a link l . The expectation of loss rate during T for a connection c given the a particular arrival sample path is the sum of the loss probabilities of individual connection packets arriving within the interval. Since the arrival sample path is fixed, the expectation is defined by the distribution of random choices of the scheduling algorithm.

The following conditions are sufficient for minimizing the maximum expected loss rates among connections over an interval of length T_0 : (1) the probability of loss for all packets taking part in an overflow event is identical, and (2) the work-conserving service rule treats all packets identically.

The first condition results from the fact that packet drop decisions are made on-line without knowledge of future packet arrivals. Since the expected loss rate of the worst connection is the sum of the loss probabilities for packets arriving within the epoch, the maximum expected loss rate given a particular arrival sample path is minimized only when the maximum loss probability in an overflow event is minimized (i.e., all participating packets are equally likely to be dropped). Otherwise, one can always construct an arrival pattern such that the connection with higher loss probability in an overflow event does not transmit further packets and thus end up with a high expected loss rate.

The second condition implies that the service rule does not prioritize packets based on their respective connections—which would result in packets from some connections spending more time in the buffer than

others and hence participating in more overflow events. Together these two necessary conditions are also sufficient since they characterize the only two policies of the scheduling algorithm.

If l employs FCFS/RD, all the packets taking part in an overflow event (i.e., present in the buffer but not in service when a new arrival triggers a buffer overflow) are subject to identical loss probability $1/(B + 1)$ due to that event, thus satisfying condition (1) above. Furthermore, as the FCFS service rule obviously satisfies condition (2), FCFS/RD minimizes the maximum expected loss rate over a single epoch given any arrival sample path. \square

3.2 Correlation of scheduling decisions

We begin by introducing a generic model of local-control scheduling algorithms and establish two properties within that model given Assumption 2 that, together with local fairness, are sufficient conditions for optimality.

Consider a link l shared by N connections $1, 2, \dots, N$. Any scheduling algorithm G at l can be viewed as assigning a loss probability $p_i^G(\tau)$ for connection i packets arriving at each time slot τ . This probability takes into account loss upon arrival and preemption from the queue if the algorithm allows. It is determined by the distribution of the joint arrival process and the distribution of upstream scheduling decision. As a concrete example, consider the Nearest-To-Go algorithm which at any link favors packets with least remaining number of hops along their paths. The probability of dropping a packet from a given connection is the probability that a certain number of packets that have fewer links to traverse are already in the buffer when the packet arrives, or arrive before the packet is served. Note that if the scheduling algorithm is randomized, the probability also depends on the distribution of algorithm's random choices.

Suppose, without loss of generality, that connections are numbered in increasing order of the loss probability at τ . That is $p_1^G(\tau) \leq p_2^G(\tau) \dots \leq p_N^G(\tau)$. Define the rank of a connection at τ as follows: The rank of connection 1, $r_1^G(\tau) = 1$ and for all j , $r_{j+1}^G(\tau) = r_j^G(\tau)$ if $p_{j+1}^G(\tau) = p_j^G(\tau)$ and $r_{j+1}^G(\tau) = r_j^G(\tau) + 1$ otherwise. Under this model it is possible that all connections have equal rank during a time slot. For example, under FCFS/RD arrivals from all connections at any slot τ have rank 1 as they are treated identically by the algorithm.

Consider an arbitrary interval $I = [t, t + nT_0)$ on link l composed of n consecutive (sub)intervals of length T_0 : $I_i = [t + (i - 1)T_0, t + iT_0)$, $i = 1, \dots, n$. Given Assumption 2, the expected loss rate of a connection is maximum if packets arrive only at those slots with the worst (highest) rank within each interval I_i . With a slight abuse of notation, let $r_c^G(i)$ be the maximum rank for connection c during interval I_i . The expected loss rate for connection c over interval I is given by:

$$\mathbf{E} \left[R_c^{l,G}(I) \right] = \frac{1}{n} \sum_{i=1}^n \mathbf{E} \left[R_c^{l,G}(I_i) \right], \quad (2)$$

where,

$$\mathbf{E} \left[R_c^{l,G}(I_i) \right] = \sum_{k=1}^N \mathbf{E} \left[R_c^{l,G}(I_i) | r_c^G(i) = k \right] \Pr \left[r_c^G(i) = k \right]. \quad (3)$$

Note that we assume that an algorithm assigns ranks to connections based on some probability distribution. Algorithms that assign ranks deterministically are a special case.

To bound the loss rate at a arbitrarily tagged connection c at link l , we consider the indicator random variables $(Z_c^{l,G}(I_1), \dots, Z_c^{l,G}(I_n))$ defined as

$$Z_c^{l,G}(I_i) \triangleq \begin{cases} 1 & \text{if } c \text{ suffers packet loss in interval } I_i \\ 0 & \text{otherwise.} \end{cases}$$

$Z_c^{l,G}(I_i)$ is a tight bound on the loss rate during subinterval i , $R_c^{l,G}(I_i)$. That is $R_c^{l,G}(I_i) \leq Z_c^{l,G}(I_i)$. In fact, $R_c^{l,G}(I_i) = Z_c^{l,G}(I_i)$ when the tagged connection offers exactly one packet to link l in the i th subinterval. Otherwise, the definition of $Z_c^{l,G}(I_i)$ assumes that any loss event experienced by the connection is subepoch i results in the loss of all packets offered by c during the subinterval.

Suppose connection c is routed along path π . Let $(Z_c^G(I_1), \dots, Z_c^G(I_n))$ be defined as $Z_c^G(I_i) \triangleq \max_{l \in \pi} Z_c^{l,G}(I_i)$. Under the assumption that G is a local-control algorithm, the distribution of $Z_c^G(I_i)$ has the following product form:⁵

$$1 - \Pr[Z_c^G(I_i) = 1] = \prod_{l \in \pi} \left(1 - \Pr[Z_c^{l,G}(I_i) = 1]\right), \quad (4)$$

Let $Z_c^G(I)$ be defined as $Z_c^G(I) \triangleq \frac{1}{n} \sum_{i=1}^n Z_c^G(I_i)$. Then $\mathbf{E}[Z_c^G(I)]$ is a tight bound on the expected loss rate for connection c over interval I . That is, $\mathbf{E}[M_c^G(I)] = \mathbf{E}[Z_c^G(I)]$. By linearity of expectations, we have

$$\mathbf{E}[Z_c^G(I)] = \frac{1}{n} \sum_{i=1}^n \mathbf{E}[Z_c^G(I_i)] = \frac{1}{n} \sum_{i=1}^n \Pr[Z_c^G(I_i) = 1] \quad (5)$$

Substituting from (4), we get

$$\mathbf{E}[M_c^G(I)] = \mathbf{E}[Z_c^G(I)] = 1 - \frac{1}{n} \sum_{i=1}^n \prod_{l \in \pi} \left(1 - \Pr[Z_c^{l,G}(I_i) = 1]\right). \quad (6)$$

Equation (6) is minimum only when the following two part condition is satisfied:

- C1:** Consider the sorted order of $\{I_1, \dots, I_n\}$ in increasing rank at some link. Equation (6) is minimum only if the sorted order is the same at every link along the path. As an example if $n = 2$ and $\mathbf{E}[Z_{c,1}^{l,G}] \leq \mathbf{E}[Z_{c,2}^{l,G}]$ at some link l along c 's path π , then $\mathbf{E}[Z_{c,1}^{l',G}] \leq \mathbf{E}[Z_{c,2}^{l',G}]$ at every $l' \in \pi$.
- C2:** The loss probability is concentrated in as few intervals as possible. That is, few intervals i have $\mathbf{E}[Z_{c,i}^G]$ much larger than $\mathbf{E}[Z_c^G]$, while the remaining ones have loss probability much smaller.

Together **C1** and **C2** are sufficient for optimality given that $G \in \text{LF}(n)$: because $\mathbf{E}[Z_c^{l,G}(I)]$ is minimum at every link l whenever $G \in \text{LF}(n)$, at any link l algorithms in $\text{LF}(n)$ can only differ in how they assign

⁵The probabilities in the product form can be conditioned on any network events, for example upstream scheduling decisions. To be precise we may write the probabilities as $\Pr[Z_c^{l,G}(I_i) = 1 | H]$ where H is the history of events in the network. We refrain from doing so for the sake of clarity.

ranks to the connection at different subintervals of I , hence in the values of $\mathbf{E}\left[Z_c^{l,G}(I_i)\right]$ for $i = 1, \dots, n$, but not in their sum—not in $\mathbf{E}\left[Z_c^{l,G}(I)\right]$.

The proof for this necessary condition proceeds by induction on the subintervals of I from 1 to n given a two hop path to show that the optimal solution has to have the loss rate concentrated in as few intervals as possible (subject to the constraint that the loss rate at any link cannot exceed 1). Then by using this result as the base case for induction on the number of hops in the path. The inductive step in both cases is straightforward, as the product-form expands into a simple inclusion-exclusion formula.

4 The Rolling Priority Algorithm

We now describe the Rolling Priority (RP) algorithm, which is parameterized with an integer $n \geq 1$. In the next section, we show that $\text{RP-}n \in \text{MML}(n)$. From the perspective of a connection, time is divided into disjoint *epochs* of fixed duration. At every link, the RP algorithm assigns scheduling priority (service and drop priority) to connections so that the packet loss probability of each connection, is high at the beginning of an epoch and quickly improves as time progresses until the start of a new epoch, at which point the process repeats. A packet retains its priority (within a small range) on all links along its path. In this section, we specify RP’s service and drop policies.

4.1 Service and drop policies

Consider a link using the RP scheduling algorithm. From the perspective of a connection, time at the link is divided into disjoint epochs of $T_e = nT_0$ time slots. Each time slot is spanned by exactly one epoch from each connection. At every time slot, RP gives scheduling priority (service and drop priority) to connections sharing the link in the order of earliest-starting current epoch, where the current epoch of a connection is the epoch spanning the time slot. The connection(s) with earliest-starting current epoch have the highest priority. Figure 1 illustrates the assignment of priority at different time slots for a link shared by three connections a, b and c . At every time slot in the interval $[t_1, t_2)$, the current epoch for connection b started earlier than the current epochs of connections a and c . As a result the priority of connection b is highest within this interval. The highest priority connection during $[t_2, t_3)$ is c , and it is a during $[t_3, t_4)$. The cycle repeats with the start of a new epoch of connection a . At the beginning of the time slot, if the number of packets available at the link (those already in the buffer and those offered by the router’s input interfaces) exceeds the buffer capacity B , the excess packets are dropped. RP drops packets from the least priority connections so that the B packets with highest connection priority remain. During the remainder of the time slot, RP serves a packet from the highest priority connection with backlog, if any.

4.2 Phase randomization and alignment of epochs

To ensure high priority packets are subject to a small loss probability at every link, RP uses randomization to avoid contention among a large number of high priority connections at any link. Furthermore, RP loosely aligns the start of connection epochs across the links it traverses so that a packet that is given high priority

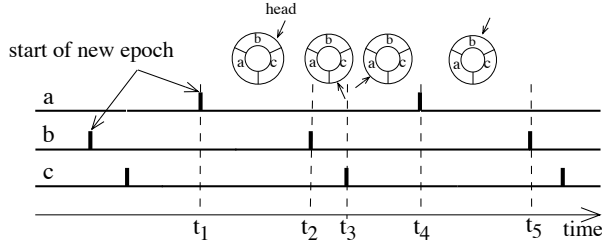


Figure 1: The priorities of three connections a, b and c at a link. The circular queue is used to enforce the cyclic priorities. The head pointer indicates the connection with highest priority at any given time.

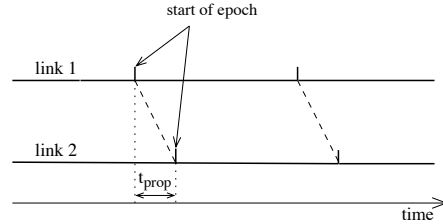


Figure 2: The start of a connection epoch at two consecutive links (link interfaces) differs by the propagation delay of the upstream link (t_{prop}).

at a link is likely to have high priority at all links along the path. Both randomization and epoch alignment are part of connection initialization that we now describe.

Each connection has an associated *phase* variable ϕ . Suppose the connection is initialized at time t_0 . The ingress router of the connection chooses the value of the phase uniformly at random from the interval $[0, T_e)$ so that the connection starts a new epoch a time $t + \phi + iT_e, i = 0, 1, 2, \dots$. The phase of the connection is communicated to downstream links in the form of one-time initialization *init* packet sent from the ingress at time $t_0 + \phi$. The reception time of an *init* packet at a given link specifies its epoch start times at that link. For instance, if an *init* packet for a particular connection is received at the link at time t , then a new epoch for the connection starts at that link at times $t + iT_e, i \geq 0$. Because RP’s service and drop policies rely on the knowledge of connection epoch boundaries, the *init* packets are always given higher scheduling priority than all data packets so that they are almost never dropped.⁶

Consider two links in tandem along the path of a connection and suppose that *init* packets do not experience any queuing delay. The start of a new connection epoch at the upstream link precedes the start of a new epoch at the downstream link by exactly the propagation delay of the upstream link (Figure 2).

4.3 Implementation of RP

The buffer capacity at a link’s interface is dynamically shared among a set virtual FIFO buffers corresponding to the set of connections sharing the link. At each time slot, the service and drop priority of each connection buffer are determined according to the earliest-starting current epoch rule. The rule can be implemented using a circular queue of pointers to the virtual connection buffers. At each time slot, the priority of a connection is the clockwise distance of the corresponding entry in the circular queue from the current position of the *head* of the queue.

RP ensures that the head pointer of the circular queue indicates the connection with earliest starting current epoch. Specifically:

⁶Initialization packets are dropped only when there are too many *init* packets at a given link, but such packets are rare since connections are traffic aggregates that are supposed to persist for long time (i.e., weeks or months). Recovery mechanisms from the loss or corruption of *init* packets is not part of the scheduling algorithm but should be provided, for example by having link interfaces notify the ingress routers of packets belonging to uninitialized connections before dropping them.

- (1) Each entry in the circular queue is associated with a counter modulo epoch duration (T_e slots). Counters are incremented at the end of every time slot. If the counter of the current head becomes zero, RP advances the head pointer clockwise to the first entry with nonzero clock.
- (2) A new entry (corresponding to an init packet from a new connection) is placed just before current head and its counter is set to zero.

The counter implementation in item (1) is conceptually the most straightforward. A more efficient implementation uses only one counter and associates with each circular queue entry a phase value equal to the reading of the counter when the entry was inserted in the circular queue. At each time step, this algorithm compares the phase of the current head to the value of the counter and advances the head pointer if they match. This implementation avoids incrementing multiple counters every time slot.

5 Analysis of Rolling Priorities

In this section we establish some properties of RP and use them to show that RP satisfies the optimality conditions of Section 3.2.

5.1 Properties of the Rolling Priority Algorithm

Let RP- n denote the algorithm RP with epoch duration $T_e = nT_0$ for some $n \geq 1$. Recall that over disjoint intervals of length T_0 at any particular link, the number of arrivals from every connection follows an identical distribution according to Assumption 2. We view each epoch as being composed of n disjoint subepochs of length T_0 slots.

We now argue that RP- n has the following properties:

Lemma 3. *Consider an epoch of a tagged connection c at some link l along its path. If the number of connections sharing the link is N , then: (i) with high probability (i.e., with probability $1 - o(1/N)$), there exist a subepoch of the epoch being considered where the scheduling priority of the connection is in the range $[\frac{N}{n}(k-1), \frac{N}{n}k]$ for $k = 1, 2, \dots, n$; and (ii) if there exist a subepoch $1 \leq i \leq n$ where the scheduling priority of the connection at link l is in the range $[\frac{N}{n}(k-1), \frac{N}{n}k]$ for some $1 \leq k \leq n$, then at every link l' along the path of the connection where the number of connection is N' , the priority of the connection during subepoch i is in the range $[\frac{N'}{n}(k-1), \frac{N'}{n}k]$ with high probability.*

Part (i) of the lemma can be explained as follows. If we consider the division of the range of priorities $[1, N]$ into n priority classes of equal size, then with high probability, $(1 - o(1/N))$, there is a subepoch where the priority of the connection is the k th priority class, for each k . Part (ii) states that if the priority of the connection in the k th priority class during a subepoch at a particular link, then, with high probability, it is in the k th priority block during the same subepoch at every link. This implies that if a packet has high (low) priority at a link, then with high probability, it has high (low) priority at every other link along the path.

Proof of Lemma 3: Consider an epoch of some arbitrarily tagged connection c at some link l along c 's path. We refer to this epoch as the *reference epoch*. The reference epoch can be viewed as composed of n consecutive subepochs of equal duration, T_0 . The subepochs are numbered $1, \dots, n$ in temporal order.

Given that the duration of an epoch is constant for all connections, the reference epoch will overlap either partially with two consecutive epochs or completely with one epoch from each connection sharing link l with c . By the random choice of phase, each connection is equally likely to start a new epoch at any time slot within the reference epoch. Without loss of generality, we consider all new epochs starting within subepoch i of the reference epoch to have started at its leading boundary.

Consider the vector (P_1, \dots, P_n) where P_i is the priority of the tagged connection during subepoch i of the reference epoch. Then P_i is a random variable representing the number of connections such that the connection's epoch overlapping with subepoch i started earlier in time than the reference epoch. It follows that P_i is stochastically dominated by the number of connections starting new epochs within the $n - i + 1$ subepochs ending with (and including) epoch 1. Due to uniform phase randomization over the duration of an epoch, this number is a binomially distributed random variable. Specifically, for $1 \leq i \leq n$, P_i is dominated by a binomial $\text{Bin}(N - 1, 1 - \frac{i-1}{n})$ random variable where we use the convention that a $\text{Bin}(N - 1, 1)$ variable deterministically assumes the value $N - 1$.⁷

Part (i) of the lemma follows from the above observation using the Chernoff bound on the tail probability of a binomial distribution (which directly gives the high probability bound). Part (ii) also follows from the same observation since the binomial approximation applies to every link along the path of the connection. \square

5.2 RP is locally fair

We now prove the following result:

Theorem 3. $\text{RP-}n \in \text{LF}(n)$ for any $n \geq 1$.

Proof. Consider an interval I of length nT_0 at a link l shared by N connection, and employing a local-control work-conserving scheduling algorithm G . Leveraging the notation of the general algorithmic model in Section 3.2, let $W_{c,i}(k)$ be an indicator variable of the event that an arbitrary connection c has rank k ($k = 1, 2, \dots, N$) during the i th subinterval I_i (of length T_0 slots) within interval I , and define $W_c(k)$ as the number of subintervals within I where the connection has rank k , i.e., $W_c(k) = \sum_{i=1}^n W_{c,i}(k)$. Then the fraction of time for which connection c assumes rank k is:

$$\mathbf{E}[W_c(k)] = \frac{1}{n} \sum_{i=1}^n \mathbf{E}[W_{c,i}(k)] = \frac{1}{n} \sum_{i=1}^n \Pr[r_c^G(i) = k], \quad (7)$$

where $r_c^G(i)$ is the rank of connection c during subinterval I_i . Suppose $G \notin \text{LF}(n)$. Then there exist a connection c such that $\mathbf{E}[W_c(k)]$ is higher than at least one other connection. Otherwise, by Assumption 2,

⁷For all i , P_i can be accurately characterized by the difference of the binomial variable and a rv representing the number of connections starting new epochs within the first subepoch (epoch 1), and whose priority tie with the tagged connection was broken by RP in favor of the tagged connection. The adopted characterization is simpler. Because the expected number of connections starting new epochs during any subepoch (thus during subepoch 1) is $(N - 1)/n$, the simple characterization is also a good approximation of P_i when n is large.

all connections must have equal expected loss rates at l which implies $G \in \text{LF}(n)$. Conversely, if $\mathbf{E}[W_c(k)]$ is constant for all connections then $G \in \text{LF}(n)$.

Now consider the algorithm $\text{RP-}n$ where interval I corresponds to the duration of an epoch and $\Pr[r_c^G(i) = k]$ corresponds to the probability distribution of priority during subepoch i of the epoch. Since the distribution of P_i was obtained for an arbitrary connection (see proof of Lemma 3), Eq. (7) indicates that $\mathbf{E}[W_c(k)]$ is constant for all connections traversing link l under $\text{RP-}n$. Thus completing the proof. \square

5.3 Optimality of Rolling Priority

We are now ready to prove our main result:

Theorem 4. $\text{RP-}n \in \text{MML}(n)$ for any $n \geq 1$.

Proof. Given that $\text{RP-}n \in \text{LF}(n)$ for every $n \geq 1$, we now argue that any work-conserving algorithm in $\text{LF}(n)$ that does not exhibit the same two properties as $\text{RP-}n$ (Lemma 3) violates either optimality conditions **C1** or **C2**, hence is not optimal.

First, suppose that an algorithm G exhibits the property in Part (i) of Lemma 3 but not the property in Part (ii). Then it obviously violates **C1**. G may not exhibit the property in Part (i) in several ways (1) it may relax the high probability condition, (2) it may allow several subintervals to assume the same class (interval) of ranks (in which case, the connection assumes ranks only in a strict subset of $\{1, \dots, N\}$), or (3) the subset of $\{1, \dots, N\}$ covered by the connection rank may not be equally subdivided among intervals.

In case (1) implies that the loss probabilities will vary less among intervals thus violating **C2**. Case (2) is similar: if the rank distribution of every connection covers a smaller set of ranks then the variation in loss probabilities among different subintervals decreases.

Case (3) is less obvious. It is clear that the loss rate during a subinterval increases with rank. The loss probability (hence rate) as a function of rank is naturally convex and increasing. This function depends on the arrival distribution and the buffer capacity at the link. Bursty arrival and/or smaller buffers imply that a larger range of ranks will exhibit high loss probability. To satisfy **C2**, an algorithm must cover the range of ranks with high loss rates in as few intervals as possible. Without knowledge of the arrival distribution and buffer capacity, any unequal division of the range $[1, N]$ into classes of ranks can be made worse than $\text{RP-}n$ by choice of arrival distribution and/or buffer size. \square

6 How far is FCFS/RD from Optimal?

In the previous section we showed that local fairness is a necessary, but not sufficient condition for optimality in the loss rate minimization problem. Specifically, an algorithm that satisfies the local fairness condition but does not satisfy the conditions on the correlation of scheduling decisions (**C1** and **C2**) is not optimal. We know only one such algorithm, namely FCFS/RD.⁸ However, it is not immediately clear how far from optimal can FCFS/RD be, and under what conditions it achieves nearly optimal loss rate bounds. This knowledge would have practical implications: when FCFS/RD is near-optimal it should be favored over

⁸Under Poisson traffic the statistical behavior of FCFS/RD and FCFS/DT are identical.

RP since it does not require connection initialization, special scheduling data structures, or per-connection virtual packet queues.

In this section, we relate the performance of FCFS/RD to the performance of the optimal algorithm RP- n . We find that under light load, any locally fair algorithm is nearly optimal. But that the difference in loss rate bounds grows quickly with load.

Consider a connection c routed along a path π of length h hops. For simplicity we assume that the joint arrival process at any link is stationary so that the time slots are indistinguishable. That is, the distribution of the total number of packet arrivals is identical at every slot. It follows that loss probability of any arriving packet is the expected loss rate at the link. Let this probability be β . Because both FCFS/RD and RP- n are work conserving the expected loss rate at any link under RP- n is also β .

Suppose for simplicity that the links have identical arrival processes. The expected loss rate of connection c under FCFS/RD over an interval $I \in \mathcal{I}_n$ is the loss probability of any of its packets. Thus, substituting into (6):

$$\begin{aligned} \mathbf{E}[M_c^{\text{FCFS/RD}}(I)] &= 1 - (1 - \beta)^h \\ &\geq 1 - e^{-h\beta} \end{aligned} \tag{8}$$

Consider the epoch of connection c corresponding to interval I and let its subepochs be numbered 1 through n . From Lemma 3, the distribution of priority during subepoch i is binomial and linearly improves with increasing i . Suppose that there exists $q \in (0, 1]$ such that the priority for all subepochs beyond subepoch $\lceil nq \rceil$ is such that the loss rate is much smaller than β , formally there exist $j : \lceil nq \rceil \leq j \leq n$ we have $\frac{\beta}{\mathbf{E}[Z_c^G(I_j)]} \geq \alpha^k$ for constants $\alpha > 0, k > 1$. Then under RP- n ,

$$\begin{aligned} M_c^{\text{RP-}n}(I) &< q + (1 - q) \left[1 - \left(1 - \frac{\beta}{\alpha^k} \right)^h \right] \\ &\approx q + (1 - q)(1 - \sqrt[k]{e^{-h\beta}}). \end{aligned} \tag{9}$$

Comparing (8) and (9), we find that when β is small (i.e., when the load at the links is light) FCFS/RD is as good as RP- n . However, under moderate-to-heavy load, (e.g., $\beta > 0.01$), $\mathbf{E}[M_c^{\text{FCFS/RD}}(I)]$ quickly approaches 1 as h increases. In contrast $M_c^{\text{RP-}n}(I)$ grows slowly with h due to the α^k root. Furthermore, when the range of h is such that $h\beta \ll \alpha^k$, $M_c^{\text{RP-}n}(I)$ saturates at q .

The difference in bounds between RP- n and FCFS/RD depends on how fast the loss rate drops across the epoch in RP- n . As an example consider the above network when traffic arrival correspond to a Poisson process. For simplicity we model each link as an M/M/1/B queue so that at load ρ the loss probability is approximately ρ^{B+1} where B is the buffer size. Under FCFS/RD (and FCFS/DT), $\beta = \rho^{B+1}$ and $\mathbf{E}[M_c^{\text{FCFS/RD}}(I)] = 1 - (1 - \rho^{B+1})^h$. Under RP- n , the load decreases linearly throughout the epoch. Thus for subepoch i have $\rho_i \approx \rho(1 - \frac{i}{n})$ and $\frac{\beta}{\mathbf{E}[Z_c^G(I_j)]} \approx \frac{\rho^{B+1}}{\rho^{B+1}(1 - \frac{i}{n})^{B+1}} = (\frac{n}{n-i})^{B+1}$, which is on the form α^k . Figure 3 is a plot comparing the performance of FCFS/RD and RP- n at different values of n when the load at the links is 90%, and $B = 5$. At $n = 10$, the loss rate nearly saturates around 0.3 and becomes

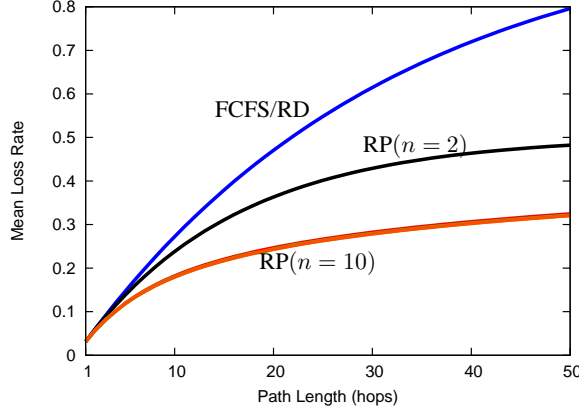


Figure 3: Numerical comparison of FCFS/RD versus RP- n under heavy load.

insensitive to the path length, indicating that beyond the third subepoch ($q = 1/3$), the loss rate at every link is negligible. For FCFS/RD, the loss rate increases to 1 with increasing path length. The RP curve for $n = 2$ highlight the role played by the length of the minimization interval. The loss rate curve saturates around 0.5 indicating that loss rate during the second subepoch is negligible is at every link.

7 Routing Tradeoffs

This section has three objectives: first, to present a method for computing tradeoffs between the maximum allowable link utilization and the maximum allowable path length so that the loss rate of every successfully routed connections satisfies a guaranteed loss rate. As we previously mentioned, these tradeoffs can be used to define constraints on connection routing (for example by fixing the path length and deriving the corresponding value for the maximum load, or vice-versa), thus are of practical value. Second, to demonstrate using numerical results that RP- n is able to achieve better tradeoffs compared to FCFS/DT (drop-tail). Finally, to demonstrate the effect of the length of the optimization interval (i.e., the number of disjoint intervals of length T_0 per epoch, n) on the utilization–path length tradeoffs.

We are interested in tradeoffs where the expected loss rate is in the order of 10^{-2} when the buffer size at every link is $B \leq 10$. This is, for instance, the case in routers with integrated optical packet buffers [5].

The contour plots in Figure 4 are obtained for RP- n with $n = 10$ assuming traffic sources that are regulated to minimize the loss rate in the network. The plots show the tradeoffs between maximum-allowable load and maximum-allowable path length at the given buffer sizes. For instance, to guarantee an expected loss rate of 0.02 when $B = 5$ at each link and the path length is 25 hops, the load at each link should not exceed 60%. The same guarantee can be supported when the maximum path length is 25 hops and $B = 10$ using a load of ≈ 0.81 .

The bounds are obtained in the following setting: each link multiplexes up to 100 periodic (CBR) connections with independent random phases, each contributing packets at a rate of $1/100$ of the link capacity. The number of connections actually sharing a link is determined by the load. The plots are upper bounds on the loss rate of a connection traversing a sequence of hops, where, at each hop, it contends with a different

set of “background” connections. It has been previously observed that given periodic sources, the case of equal bandwidth allocations results in highest loss rate at every link [27].

The bounds are obtained using the overflow probability for $ND/D/1$ queues in [27] using the product-form of Eq. (4). Note that given the same arrival process, all work-conserving algorithms have equal loss rate (overflow probability). Here, the product form assuming link independence yields a worst-case bound since it captures the case where (1) the connection contends at every hop with a different set of connections, and (2) the load due to background connections at every hop has not been thinned by packet loss at upstream links. In contrast, if a connection contends with the same set of connections at every link, contention among packets is resolved at the first bottleneck, and no loss occurs at subsequent links.

Figure 5(a) shows the tradeoffs supported by FCFS/DT at a buffer size $B = 5$. Figures 5(b) and 5(c) show the tradeoffs for RP- n at $n = 5$ and $n = 20$, resp., given the same values for B . Comparing Figures 5(b) and 5(c), we find that larger values of n yield better tradeoffs, as we expect based on the discussion in Section 6. On the other hand, comparing Figures 5(a) and 5(c), shows that RP- n improves the tradeoffs compared to FCFS/DT. For instance, whereas FCFS/DT cannot satisfy a loss rate bound of 0.02 over paths longer than 8 hops at 60% link utilization, RP- n with $n = 10$ and above, can provide the same guarantee over 25-hop paths (from Figure 4(a)). That is, RP- n allows the network to route traffic over paths that are unusable under FCFS/DT.

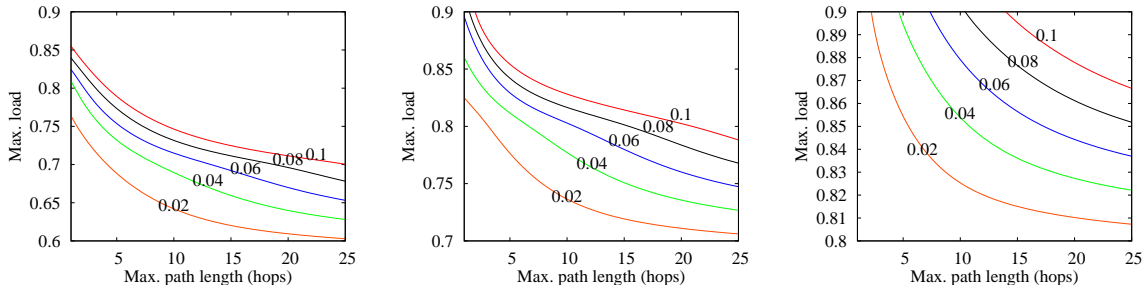


Figure 4: Path length vs. utilization tradeoffs at constant loss rate using RP- n with $n = 10$. Left: (a) $B = 5$, middle: (b) $B = 7$, right: (c) $B = 10$.

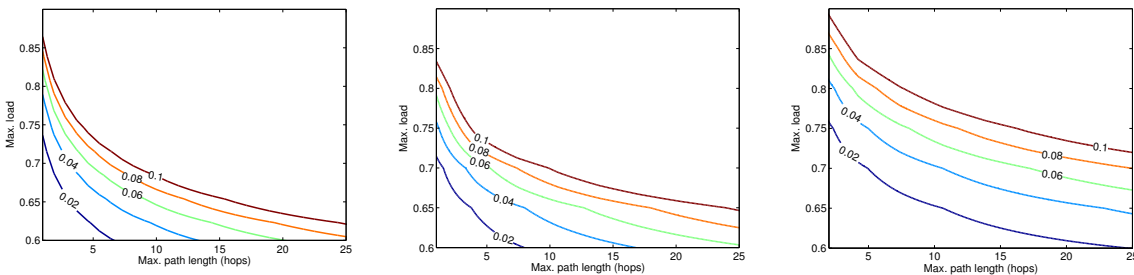


Figure 5: Path length vs. utilization tradeoffs at constant loss rate and $B = 5$, Left: (a) FCFS/DT, middle: (b) RP- n with $n = 5$, right: (c) RP- n with $n = 20$.

8 Simulation-based Comparison of Scheduling Algorithms

In this section, we report simulation experiments that compare the observed loss rate under RP- n , FCFS/DT (drop-tail), FCFS/RD, and a buffer-constrained implementation of Weighted Fair Queuing. The simulation experiments are used to support the analytical results presented so far by giving examples of the observed performance under different algorithms.

8.1 Comparison to FCFS/DT and FCFS/RD packet scheduling

Here, we compare the observed tradeoff between load and path length to achieve a desired loss rate under RP to the tradeoff observed in a network using FCFS/DT. The results show that there is at least a case where the difference in the numerical bounds translates into actual difference in observed performance.

To differentiate the performance of RP and FCFS/DT, we used a “parking-lot” topology (Figure 6), where a tagged (foreground) connection traverses a path of identical links. At each link, the tagged connection competes with a different set of background connections. These connections do not face any contention, except at the link shared with the tagged connection. As in the previous section, all network connections are periodic to emulate ingress-shaped traffic and have identical bandwidth allocations, equal to $1/100$ of link bandwidth. All the links along the path have equal loads, and thus are shared by the same number of connections.

We conducted experiments using *ns2* [1] simulator at different values of epoch lengths (n), link loads, and buffer sizes. Figure 7(c) is a contour plot of the observed tradeoff between load and path length to achieve a desired average loss rate. In these experiments, the buffer size was set to 5 packets at every link and the length of the RP epoch was set to $n = 10$. The plot was obtained by running a set of 100 experiments for each (load, path-length) pair and averaging the loss rate of the tagged connection. Similar experiments are reported for FCFS/DT (Figure 7(a)) and FCFS/RD (Figure 7(b)).

Comparing Figures 7(a) and 7(b) we find that the average packet loss rate of FSFS/DT and FSFS/RD is similar under moderate load conditions when the maximum path length (in terms of hops) is relatively short. As the path length increases beyond a few hops, the difference between the loss rate of the two algorithms increases, with FCFS/RD giving a slightly better average performance. This decrease in loss rate for FCFS/RD can be attributed to the local fairness property as discussed earlier in §3.

However, we find that both FCFS/DT and FCFS/RD are much more sensitive to path length compared to the RP (Figure 7(c)). In particular, whereas a loss rate of 0.02 can be maintained for up to 20 hops at 60% load under the proposed scheme, it can only be maintained for only for less than 5 hops using FCFS/DT. Observe that the tradeoff under FCFS/DT is slightly worse than the bounds obtained in the previous section. This is due to allowing the periodic sources to randomly perturb the interval between consecutive packets in order to avoid traffic phase effects that may lead to the starvation of some connections (all packet transmissions from those connections are lost). Traffic phase effects can occur as well in real networks if traffic is shaped into periodic streams.⁹ In practical settings however, shapers (such as leaky buckets) allow a degree of burstiness that is similar to the random perturbation. The simulation results are

⁹See for example [17]

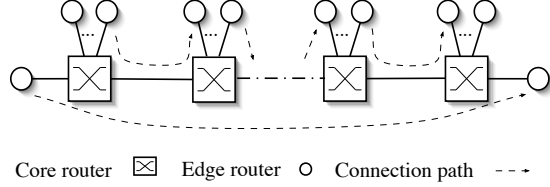


Figure 6: Simulation topology

therefore more representative of the performance of FCFS/DT and FCFS/RD than the analytical bounds.

8.2 Randomized Traffic performance

In this section, we compare the performance of FCFS/DT, FCFS/RD, and a Buffer-Constrained implementation of Fair Queuing (BCFQ) under conditions where the traffic sources show significant random behavior compared to the simulations described earlier. This increased randomness is introduced in the packet inter-arrival times of consecutive packets from the same source, so that the inter-arrival times range uniformly from $(0, \tau)$, where τ is the inter-arrival time calculated without any perturbation. Other than the traffic characteristics, the same topology as described in earlier section was used in these simulations. The increased randomness is introduced to break packet transmission synchronization among competing connections, hence reducing the benefit that RP and FCFS/RD have due to the local fairness property.

BCFQ was implemented by modifying the Fair Queuing module in *ns2*. The standard Fair Queuing algorithm maintains a separate queue for each connection and, in our setting where packets have fixed size, it serves each queue in a round-robin fashion. Thus a source that is trying to gain more than its fair share of capacity will merely increase the occupancy of its own queue. In BCFQ, we bounded the total buffer occupancy (over all the existing queues). In this case, an arriving packet will only get buffered if the total buffer occupancy is less than a certain bound B . Fairness in BCFQ refers to fairness in service rates offered to the different queues, as opposed to the fairness in apportioning loss among connections in FCFS/RD and RP. The drop policy for BCFQ is the drop tail policy: an incoming packet is dropped if it arrives to a full queue (total buffer occupancy is B).

The simulation results for FCFS/DT, FCFS/RD, and BCFQ are given in Figure 8 for the case of $B = 5$. The results show similar performance for all three algorithms; with FCFQ/DT giving slightly higher average loss rate compared to the other two algorithms. This similarity in performance can be explained by the decrease in synchronization in packet transmission from different flows. Still the improved performance does not match the tradeoffs observed under RP in Figure 7(c). This shows that the superiority of RP with respect to loss guarantees translates into actual difference in observed performance.

9 Concluding Remarks

Motivated by the goal of efficiently providing loss guarantees in packet networks we have formulated a problem of scheduling to minimize a bound on the expected loss rate of every network connection (aggregate

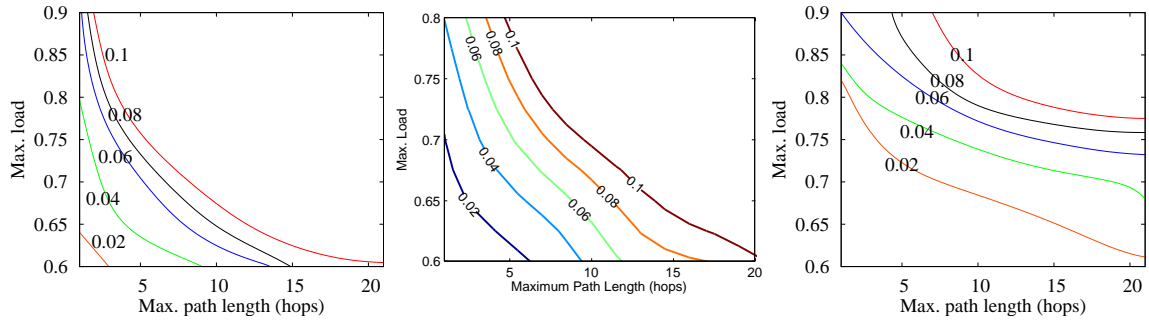


Figure 7: Observed tradeoff between load and path length to achieve a desired loss rate when $B = 5$. Left: (a) FCFS/DT, middle: (b) FCFS/RD, and right: (c) RP ($n = 10$).

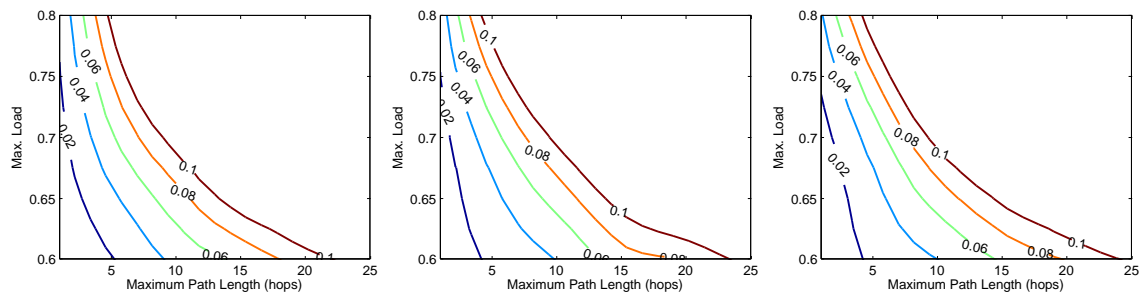


Figure 8: Observed tradeoff under randomized traffic and $B = 5$. Left: (a) FCFS/DT, middle: (b) FCFS/RD, and right: (c) BCFQ.

flow). We identified necessary and sufficient conditions for optimality. Specifically, we found that an optimal algorithm must satisfy a local fairness condition whereby it ensures that the maximum expected loss rate among connections sharing each link is minimized. In addition, the scheduling decisions at a link and along the path must be statistically correlated to ensure that packets receive consistent treatment at every hop.

We showed that the algorithm combining the FCFS service rule with the random drop policy (FCFS/RD) is locally fair but does not satisfy the correlation conditions. We then introduced a novel work-conserving algorithm called Rolling Priority (RP), that is designed to satisfy the local fairness and correlation conditions and established its optimality. RP relies exclusively on local information and does not employ explicit coordination of scheduling decisions at different hops.

One limitation of the RP algorithm is that it encourages loss to occur in bursts, which may be harmful to some applications. Therefore, one open question is whether there is an algorithm that does not suffer from this limitation and that is also (nearly) optimal. There are several other avenues for future research in networks with limited buffers. One limitation area that is receiving increasing attention is that of throughput-competitive scheduling in packet networks with fixed-size buffers—under adversarial traffic injection [2]. Algorithms shown not to suffer from throughput collapse under such adversarial conditions are all based on fixed priority, for example the Nearest-To-Go algorithm, hence not optimal for the minimization problem tackled in this paper. One interesting question is whether the Rolling Priority algorithm is also throughput-competitive under adversarial traffic.

References

- [1] The network simulator ns-2. <http://www.isi.edu/nsnam/ns/>.
- [2] W. Aiello, R. Ostrovesky, E. Kushilevitz, and A. Rosén. Dynamic routing on networks with fixed-size buffers. In *Symposium On Discrete Algorithms (SODA)*, 2003.
- [3] Matthew Andrews. Instability of FIFO in session-oriented networks. *J. Algorithms*, 50(2):232–245, 2004.
- [4] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *ACM SIGCOMM '04*, August /September 2004.
- [5] Neda Beheshti, Yashar Ganjali, Ramesh Rajaduray, Daniel Blumenthal, and Nick McKeown. Buffer sizing in all-optical packet switches. In *Optical Fiber Communication (OFC)*, 2006.
- [6] Allan Borodin, Jon Kleinberg, Prabhakar Raghavan, Madhu Sudan, and David P. Williamson. Adversarial queuing theory. *J. ACM*, 48(1):13–38, 2001.
- [7] Cheng-Shang Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, London, UK, 2000.
- [8] Cheng-Shang Chang, Yi-Ting Chen, and Duan-Shin Lee. Constructions of optical FIFO queues. *IEEE/ACM Trans. Netw.*, 14(SI):2838–2843, 2006.
- [9] Shang-Tse Chuang, Ashish Goel, Nick McKeown, and Balaji Prabhakar. Matching output queueing with a combined input-output queued switch. In *IEEE INFOCOM*, 1999.
- [10] J. A. Cobb, M. G. Gouda, and A. Elnahas. Time-shift scheduling: Fair scheduling of flows in high-speed networks. *IEEE/ACM Transactions on Networking*, 6(3):274–285, June 1998.
- [11] Amogh Dhamdhere, Hao Jiang, and Constantinos Dovrolis. Buffer sizing for congested internet links. In *IEEE INFOCOM*, 2005.
- [12] M. Elhaddad, R. Melhem, and T. Znati. Analysis of a transmission scheduling algorithm for supporting bandwidth guarantees in bufferless networks. *ACM Sigmetrics Performance Evaluation Review*, December 2006.

- [13] M. Elhaddad, R. Melhem, and T. Znati. Supporting loss guarantees in buffer-limited networks. *International Workshop on Quality of Service (IWQOS)*, June 2006.
- [14] M. Enachescu, Y. Ganjali, A. Goel, N. McKewon, and T. Roughgarden. Routers with very small buffers. In *IEEE Infocom*, 2006.
- [15] Sally Floyd. Proposed modifications to RED, and other proposals for active queue management. [Online:] <http://www.icir.org/floyd/red.html>. (A list of AQM proposals).
- [16] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1(4):397–413, 1993.
- [17] Sally Floyd and Van Jacobson. Traffic phase effects in packet-switched gateways. *Journal of InterNetworking: Practice and Experience*, 3(3):115–156, September, 1992.
- [18] E. Gordon and A. Rosén. Competitive weighted throughput analysis of greedy protocols on DAGs. In *PODC '05: Proceedings of the twenty-fourth annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 227–236, New York, NY, USA, 2005. ACM Press.
- [19] Mor Harchol-Balter and David Wolfe. Bounding delays in packet-routing networks. In *the 27th Annual ACM Symposium on Theory of Computing (STOC)*, pages 248–257, May 1995.
- [20] J.-Y. Le Boudec and P. Thiran. *Network Calculus: A theory of deterministic queues for the Internet*. Number 2050 in LNCS. Springer Verlag, 2002.
- [21] Emilio Leonardi, Marco Mellia, Marco Ajmone Marsan, and Fabio Neri. Joint optimal scheduling and routing for maximum network throughput. 2005.
- [22] C. Li and E. Knightly. Coordinated multihop scheduling: A framework for end-to-end services. *IEEE/ACM Trans. Netw.*, 10(6), December 2002.
- [23] M. May, J. Bolot, C. Diot, and B. Lyles. Reasons not to deploy RED. In *Proc. of 7th. International Workshop on Quality of Service (IWQoS'99), London*, pages 260–262, June 1999.
- [24] N. McKeown and D. Wischik. Hot Topic: Making router buffers much smaller. *SIGCOMM Comput. Commun. Rev.*, 35(3):73–74, 2005.
- [25] Abhay K. Parekh and Robert G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Trans. Netw.*, 2(2):137–150, 1994.
- [26] M. Reisslein, K. W. Ross, and S. Rajagopal. A framework for guaranteeing statistical QoS. *IEEE/ACM Trans. Netw.*, 10(1):27–42, 2002.
- [27] J. W. Roberts and J. T. Virtamo. The superposition of periodic cell arrival streams in an ATM multiplexer. *IEEE Trans. Commun.*, 39(2):298–303, Feb. 1991.
- [28] Taieb Znati and Rami G. Melhem. Node delay assignment strategies to support end-to-end delay requirements in heterogeneous networks. *IEEE/ACM Trans. Netw.*, 12(5):879–892, 2004.