Paging Against a Distribution and IP Networking

Carsten Lund Steven Phillips Nick Reingold

AT&T Bell Laboratories
600 Mountain Ave.
Murray Hill, NJ 07974

Abstract

In this paper we consider the paging problem when the page request sequence is drawn from a distribution, and give an application to computer networking.

In the IP-paging problem *the page inter-request times are chosen according to independent distributions. For this model we construct a very simple deterministic algorithm whose page fault rate is at most 5 times that of the best online algorithm (that knows the inter-request time distributions). We also show that many other natural algorithms for this problem do not have constant competitive ratio.*

In distributional paging *the inter-request time distributions may be dependent, and hence any probabilistic model of page request sequences can be represented. We construct a simple randomized algorithm whose page fault rate is at most 4 times that of the best online algorithm.*

The IP-paging problem is motivation by the following application to data networks. Next generation wide area networks are very likely to use connection-oriented protocols such as Asynchronous Transfer Mode (ATM). For the existing investment in current IP networks such as the Internet to remain useful, we must devise mechanisms to carry IP traffic over connection-oriented networks. A basic issue is to devise holding policies for virtual circuits carrying datagrams; for some connection-oriented networks the holding policy problem is exactly IP-paging.

1 Introduction

This paper studies IP-paging, a problem that can arise when carrying IP traffic over a connection-oriented network, as well as distributional paging, the most general possible randomized model of request sequences for the paging problem.

We give simple and efficient algorithms for these problems, and prove that both algorithms achieve within a constant factor of the best possible fault rate.

1.1 Motivation: carrying IP traffic over connection-oriented networks

In recent years there has been a rapid proliferation of computer networks. At the same time there has been a variety of networking standards organizations (CCITT, ATM Forum, IEEE, ANSI) developing networking protocols (Broadband ISDN / ATM, FDDI, DQDB, SMDS, Frame Relay, SONET etc.) The movement towards inter-networking has resulted in a Tower of Babel, in which networks using a myriad different protocols must interconnect. There are many practical and theoretical problems involved in efficiently interfacing networks with different protocols.

Currently the Internet is the world's largest computer network, connecting more than a million computers. The Internet uses the Internet Protocol (IP), which is *connectionless* — communication is modeled on a postal network. Injecting a packet into the network is analogous to mailing a letter. However, there is an ongoing rapid deployment of new high-bandwidth protocols such as Frame Relay and Asynchronous Transfer Mode (ATM), that are *connection-oriented*, like a phone network. For a pair of hosts to communicate, a *virtual circuit* must first be set up, consisting of a route for data in the conversation.

Because of the huge installed base of Internet hosts and the imminent widespread availability of connection-oriented networks, it is important to study means of efficiently carrying Internet traffic over connection-oriented networks.

1.2 The IP-paging problem

We consider an IP host transmitting data over a connection-oriented network. The IP host receives packets (or datagrams) that it must send to a variety of destinations in the network. The arrival of an IP packet should cause a virtual circuit to be opened, if one is not open already. The problem we study is the efficient management of virtual circuits at the IP host.

The measure of efficiency of virtual circuit management depends heavily on the pricing policy of the connection-oriented network. Some virtual circuit oriented networks treat virtual circuits as a valuable resource, and limit the number of virtual circuits that a host may have open simultaneously. This is often true of X.25 networks [11], where the limit is typically between 32 and 128. We are concerned with such networks, and use the following pricing scheme, considered in [12, 2, 10, 8]: the host pays a fixed charge each time a virtual circuit is set up. We note that there are many other possible pricing policies for connection-oriented networks and, in particular, future ATM networks are likely to have more complicated pricing policies depending on usage and quality of service.

Let k denote the maximum number of virtual circuits that a particular host may have simultaneously open. Because there is no charge for keeping a virtual circuit, the host should keep k virtual circuits open all the time. Whenever a packet arrives at the host, if there is no virtual circuit open to the destination, the host must choose some virtual circuit to close, before opening a circuit for the packet.

A *conversation* refers to communication between a pair of hosts. We refer to a conversation as "open" if it currently has a virtual circuit assigned to it. By "closing a conversation" we mean closing the corresponding virtual circuit.

We make the following basic assumptions:

1. Packet arrival times for different conversations are independent.

2. Each conversation has a fixed *inter-arrival time distribution*. After each packet in the conversation, the inter-arrival time (until the next packet arrives) is drawn from the same distribution.

3. The inter-arrival time distributions are known to the algorithm managing the virtual circuits.

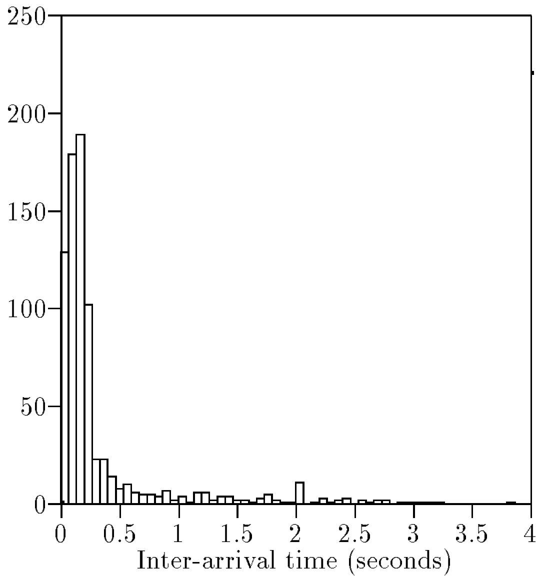

Figure 1: Inter-arrival time distributions for a bursty conversation.

We call the problem of deciding which circuit to close, under these assumptions, the *IP-paging problem*.

Assumption 1 should not be controversial, and is often made in the study of computer networks. Assumption 3 can be approximated in practice by learning the distribution from observations. Assumption 2 is stronger, and asserts that conversations have coherence through time. For example, a bursty conversation (such as a file transfer) remains bursty, while a conversation with fairly regular inter-arrival times (such as a user typing in a remote login session) remains regular. Our algorithms do not actually need assumption 2, instead it suffices to know the current distribution at any time. However in practice, if the distributions are to be learned, it is necessary for them to remain consistent through time.

We have performed an experimental study [10, 8] showing that an implementation of the algorithm presented in Section 2, based on these three assumptions (and learning the distributions) outperform previously known heuristics (such as Least Recently Used) on packet traces obtained from actual networks.

An inter-arrival time distribution that was observed in the experimental study is shown in Figure 1. This conversation is bursty, consisting of groups of packets arriving in quick succession, with bursts separated by gaps of various sizes.

For ease of presentation we will use discrete time distributions rather than continuous time. If this ever results in two packets arriving simultaneously on different conversations, we assume that they arrive in random order.

It is important to note that the distribution of the length of time until the next packet arrives in a conversation depends on the length of time since the last packet. In the conversation of Figure 1, if a packet has just arrived, the next packet is likely to arrive very soon. However, if a second has passed since the last packet, the next packet will probably take a while, as the conversation is between bursts.

1.3 Distributional paging and the best online algorithm (ON)

IP-paging is clearly a close relative of paging in a two-level storage system, where a paging algorithm must decide which page to evict on a page fault. Virtual circuits correspond to page slots in fast memory. However, there is a fundamental difference: references to different pages are certainly not independent, so page request sequences do not satisfy the above assumption 1.

The most general probabilistic model of paging is *distributional paging*, in which there is a completely unrestricted distribution over page request sequences. Distributional paging is more general than IP-paging, since assumptions 1 and 2 need not hold. It also generalizes and subsumes previous probabilistic models of page request sequences, including Markov paging [7] and the independent page model (see for example [5]).

Let ON denote the best online algorithm for IP-paging or distributional paging, depending on context: ON makes optimal use of knowledge of the packet inter-arrival time distributions, but does not have prior knowledge of the actual sequence of packet arrival times. Thus ON will typically have a higher expected cost than the optimal *offline* algorithm, and in the worst case, ON has a cost which is a factor of $\theta(\log k)$ greater than the optimal offline cost.

Given discrete inter-arrival time distributions for some conversations, IP-paging can be described as a Markov decision problem, in which a state describes which conversations are open and the length of time since the last packet in each conversation. It follows from Markov decision theory [3] that ON may be determined by solving a linear program. Unfortunately, the state space and hence the linear program have size exponential in the number of conversations and the descriptions of the inter-arrival time distributions. It is not known whether ON can be found in polynomial time, or even whether ON has a polynomial size description that can be used to service each packet in polynomial time. Hence we are concerned with algorithms that approximate the performance of ON. It remains an interesting open question to prove a lower bound for determining ON.

Definition 1.1 An algorithm $\mathcal{A}$ is said to be *c-competitive against* ON if given any starting configuration $\mathcal{C}$ (consisting of a list of k conversations that are initially open, and the time since the last packet for each conversation), and for each value of n, the expected number of virtual circuits set up by $\mathcal{A}$ starting from configuration $\mathcal{C}$ and ending at time n is at most c times the expected number of virtual circuits set up by ON starting from configuration $\mathcal{C}$ and ending at time n.

The competitive ratio is often defined by a limiting ratio, which allows algorithm $\mathcal{A}$ an extra additive constant cost. The definition used here gives a stronger bound on the cost of $\mathcal{A}$. Our lower bounds are also valid when an additive constant cost is allowed.

1.4 Results

We present a simple algorithm for IP-paging, the median algorithm, which always closes the virtual circuit for which the median of the distribution on the remaining time until the next packet is largest.

Our first main result is a proof that the median algorithm is no more than 5-competitive against ON. The result is proved in Section 2. If the algorithm does not initially know the distributions, but rather learns by observation, then this result is true only for the limiting ratio definition of competitive ratio, since the algorithm will incur additional costs during its initial learning phase.

Section 3 presents the second main result. We use randomization to improve on the median algorithm by obtaining an algorithm that is 4-competitive against On. Much more importantly though, we do not need the above assumptions 1 and 2, so the result holds in the general distributional paging model. The algorithm presented here is simpler and achieves a better competitive ratio against On than the less general Markov paging algorithm described in [7]. We make no assumption about the distribution on page request sequences, instead we assume only that the paging algorithm can determine, for each pair p, q of pages in fast memory, the probability that p will next be referenced before q.

To show that this information suffices for a 4-competitive algorithm, we introduce the notion of a dominating distribution in a tournament, and prove that every tournament has a dominating distribution.

Section 4 shows that the median is the optimal parameter of the distribution for a deterministic algorithm to use, given our method of proof of competitiveness. This optimality can be expressed as the following result about probability distributions over the reals. If f is any function that maps distributions over $\mathcal{R}$ to real numbers, then for any $\epsilon > 0$ there exist distributions $\mathcal{P}$ and $\mathcal{Q}$ over $\mathcal{R}$ such that $f(\mathcal{P}) \leq f(\mathcal{Q})$, but if p and q are independent random samples drawn from $\mathcal{P}$ and $\mathcal{Q}$, then $\Pr[p \leq q] \leq 1/4 + \epsilon$. Intuitively this result bounds how well f can predict $\Pr[p \leq q]$.

Section 5 is devoted to lower bounds. First some natural algorithms are shown not to have constant competitive ratio: the algorithm that closes the virtual circuit for which the expected time until the next packet is largest, the algorithm that closes the virtual circuit that is most likely to be last, and the Least Recently Used algorithm. Then a lower bound on the competitive ratio of the median algorithm is presented. By extensive computer search, using linear programming to find the optimal algorithm for each set of distributions, the best lower bound we have found is $1.511\cdots$, but for intelligibility, we present a simpler lower bound of 1.4.

1.5 Related Work

The problem of managing virtual circuits when carrying IP packets over connection based networks has been previously considered in [2] and [12]. Caceres [2] describes the implementation of an IP over X.25 interface. Saran and Keshav [12] performed an empirical study of several policies for managing virtual circuits in IP-over-ATM networks. Together with Keshav and Saran, the present authors [10, 8] performed an empirical study of adaptive policies for virtual circuit management, based on the the optimal use of inter-arrival time distributions. They examined two network pricing policies, of which one was IP-paging. The study showed that an implementation of the median algorithm significantly outperforms other known algorithms for IP-paging on real data gathered from networks, even when including the initial costs incurred by the median algorithm while learning the distributions. The improved performance comes at the cost of increased complexity, due to maintenance of the learned distributions, and it is left as a case-specific engineering decision whether the improvement justifies the increased complexity. The reader is referred to the article [8] for more details.

The paging problem has received much attention as the archetypal problem in online algorithms (see for example [14, 4, 1, 6] and references therein). Restricted cases of distributional paging (where only restricted distributions on page request sequences are allowed) also have a long history. Early paging work included a model in which at each time step, the page that is requested is drawn independently from a fixed probability distribution over the set of pages (see for example Franaszek and Wagner [5]). Shedler

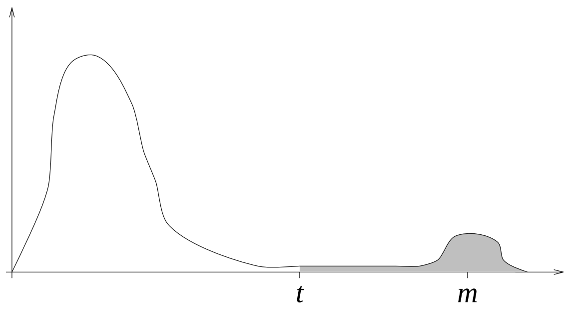

Figure 2: The median m conditioned on the fact that the last packet arrived t time steps ago.

and Tung [13] and Lewis and Shedler [9] study the behavior of the *least recently used* (LRU) paging algorithm when page request sequences are generated by a Markov chain whose states represent *LRU stack distances*. Karlin, the second author, and Raghavan [7] study the Markov paging problem, in which page request sequences are generated by a Markov chain whose states are the pages of memory, one state per page.

Because distributional paging assumes nothing about the distribution on page request sequences, we do not concern ourselves with the large literature on models and properties of page request sequences.

2 The Median Algorithm

In this section we present and analyze the median algorithm for the IP-paging problem.

Definition 2.1 The *median* of a distribution over $\mathcal{R}^+$ is the least value of t such that at least half the distribution is at most t. Formally, if X is a random variable drawn from the distribution, then the median is the least t such that $\Pr[X \leq t] \geq 1/2$.

Definition 2.2 When a packet arrives on a conversation that does not have an open circuit, the *median algorithm* closes the virtual circuit with the largest median time until the next packet.

Some observations are in order. Firstly, the median algorithm uses only a limited amount of the information available to it. Secondly, at the time when a closed conversation has a packet, the median time until the next packet for a conversation C is *not* just the median of its inter-arrival time distribution. Instead, it is the median of a *conditional distribution*, the inter-arrival time distribution of C conditioned on the inter-arrival time being at least the elapsed time since the previous packet in C, minus the elapsed time. See Figure 2.

The fact that the median algorithm is competitive follows easily from the following lemma. Informally, the lemma shows that if each choice that an IP-paging algorithm makes (of which conversation to close) is good in a local sense, then the algorithm has good global performance.

Lemma 2.3 *Let $\mathcal{A}$ be a deterministic IP-paging algorithm. Assume that at each time t that $\mathcal{A}$ closes a conversation p, the following property holds: for every open conversation*

q, the probability that q has its next packet no later than p is at least $1/c$. Then $\mathcal{A}$ is $(c+1)$-competitive against ON.

Proof We use a charging scheme, where each time $\mathcal{A}$ closes a conversation p, p places a charge on a conversation $c(p)$ that ON has closed, that is likely to receive a packet no later than p. Hence ON is likely to have to reopen $c(p)$ no later than $\mathcal{A}$ reopens p. We need only be careful that no conversation can receive many charges.

Let s be the conversation that leads $\mathcal{A}$ to drop the conversation p. Let ON^+ be the set of conversations that ON has open after the packet from s and let $\mathcal{A}^-$ be the set of conversations that $\mathcal{A}$ has open before the packet from s. Note that $s \in \text{ON}^+ \setminus \mathcal{A}^-$. When p is dropped, the conversation $c(p)$ is chosen as follows:

1. If $p \notin \text{ON}^+$ then set $c(p) = p$.
2. Otherwise, as we shall see later, there is some conversation q in $\mathcal{A}^- \setminus \text{ON}^+$ that has received no charge from $\text{ON}^+ \setminus \mathcal{A}^-$. Set $c(p) = q$.

The charge is *dropped* the next time p has a packet, and is *open* until then. Lastly, if p is dropped by ON while it has an open charge on some conversation, we reset $c(p)$ to p.

These rules imply that at any time if $c(p) \neq p$ then $p \in \text{ON} \setminus \mathcal{A}$. Furthermore note that when p is dropped then $s \in \text{ON}^+ \setminus \mathcal{A}^-$ and since s just had a packet, s has no open charge on any conversation. Thus some $q \in \mathcal{A}^- \setminus \text{ON}^+$ has received no charge from $\text{ON}^+ \setminus \mathcal{A}^-$ since $|\text{ON}^+ \setminus \mathcal{A}^-| = |\mathcal{A}^- \setminus \text{ON}^+|$.

Notice that each conversation has at most two charges on it at any time. In addition, with probability at least $1/c$, $c(p)$ has its next packet no later than p. If $c(p) = p$ this follows from the assumption about $\mathcal{A}$, since $c(p)$ is open when p is closed. The change in $c(p)$ when ON closes p can only increase the probability that $c(p)$ has its next packet no later than p.

To complete the proof, consider an infinite sequence of packet arrivals, and fix some time n. We compare the expected number of circuits closed by $\mathcal{A}$ and ON until time n. Let $r(t,p)$ be the next time p has a packet after time t. The following indicator variables are useful:

- $\chi(t,p)$ is 1 iff $\mathcal{A}$ closes conversation p at time t.
- $\alpha(t,p)$ is 1 iff at time t, $\mathcal{A}$ closes p and $c(p) \neq p$, and in addition $c(p)$ has a packet by time $r(t,p)$.
- $\beta(t,p)$ is 1 iff at time t, $\mathcal{A}$ closes p and $c(p) = p$.

Lastly let $\mathcal{O}$ be the number of open charges at time n, and let $\mathcal{I}$ be the number of conversations that are not open at time 0, but have a packet by time n.

The number of circuits closed by $\mathcal{A}$, $C_{\mathcal{A}}$, is given by

$$C_{\mathcal{A}} = \sum_p \sum_{t<n} \chi(t,p),$$

while C_{ON}, the number of circuits closed by ON, satisfies the following two inequalities

$$C_{\text{ON}} \geq \sum_p \sum_{t<n} \beta(t,p) + \mathcal{I} - \mathcal{O}. \tag{1}$$

$$C_{\text{ON}} \geq \sum_p \sum_{t<n} \alpha(t,p) + \mathcal{I} - \mathcal{O}. \tag{2}$$

Adding (1) to c times (2) gives

$$\begin{aligned}(c+1)C_{\text{ON}} \;\geq\;& c\cdot\sum_p\sum_{t<n}\alpha(t,p)+\sum_p\sum_{t<n}\beta(t,p)\\ &+(c+1)\mathcal{I}-(c+1)\mathcal{O}.\end{aligned}$$

Now by assumption we have $E[c\cdot\alpha(t,p)+\beta(t,p)] \geq E[\chi(t,p)]$, from which we obtain

$$(c+1)E[C_{\text{ON}}] \;\geq\; E[C_{\mathcal{A}}]+(c+1)\cdot E[\mathcal{I}-\mathcal{O}] \;\geq\; E[C_{\mathcal{A}}].$$

The last inequality follows from the observation that any conversation that is currently held open by $\mathcal{A}$ or is giving a charge must have been initially open, or it must have had a packet by time n. In symbols, $k+\mathcal{O}\leq k+\mathcal{I}$, whence $\mathcal{I}\geq\mathcal{O}$. ∎

Theorem 2.4 *The median algorithm is 5-competitive against* ON.

Proof Consider a time t when the median algorithm closes a conversation p, and let q be any other open conversation at time t. With probability at least $1/2$, q has its next packet no later than its median. Similarly, with probability at least $1/2$, p has its next packet no sooner than its median. By the independence of the inter-arrival time of different conversations, the probability that the next packet in q is no later than the next packet in p is at least $1/4$. Hence by Lemma 2.3, the median algorithm is 5-competitive against ON. ∎

Note that the above proof uses no information about the distributions. For many distributions, the competitive would be better than 5. In fact we have not been able to find a lower bound of more than 1.511 for any set of distributions, see Section 5.2.

The proof of Lemma 2.3 uses the fact that $\mathcal{A}$ is deterministic. When $\mathcal{A}$ is randomized the best we can prove is $2c$-competitiveness.

Lemma 2.5 *Let $\mathcal{A}$ be an IP-paging algorithm. Assume that at each time t that $\mathcal{A}$ closes a conversation p chosen from some distribution, the following property holds: for every open conversation q, the probability that q has its next packet no later than p is at least $1/c$. Then $\mathcal{A}$ is $2c$-competitive against* ON.

Proof We will have only indicator variable $\gamma(t,p)$ which is 1 iff at time t, $\mathcal{A}$ closes p and $c(p)$ has a packet by time $r(t,p)$. Now the following inequalities are satisfied:

$$C_{\text{ON}} \geq \frac{1}{2}\sum_p\sum_{t<n}\gamma(t,p)+\mathcal{I}-\frac{1}{2}\mathcal{O} \tag{3}$$

$$E[c\cdot\gamma(t,p)] \geq E[\chi(t,p)], \tag{4}$$

where the expectation is taken over both future requests and the distribution that $\mathcal{A}$ uses to choose p. The division by 2 in (3) is because each conversation that ON closes might pay two charges. For two conversations a and b, let "$a \ll b$" denote the event that the next packet from a arrives no later then the next packet from b. To show (4) note that $c(p)$ is either p in which case $p \ll c(p)$ always or $c(p)$ is some fixed $q\in\mathcal{A}\setminus$ ON that has no current charges. Thus $\Pr[p\ll c(p)] \geq \Pr[p \ll q] \geq 1/c$. This implies (4). The rest of the proof is similar to the proof of Lemma 2.3. ∎

3 Paging against an arbitrary distribution

In this section we consider the problem of paging when the page request sequence is drawn from a completely arbitrary distribution $\mathcal{D}$. For brevity we call this problem *distributional paging*.

Distributional paging is more general than the IP-paging problem, since the inter-arrival times of different conversations are not assumed to be independent and the inter-arrival time distribution of a conversation may vary over time. It also subsumes the Markov paging problem [7], in which the page request sequence is generated by a Markov chain whose states are the pages of memory, one state per page. Distributional paging is more general than Markov paging, since there are simple distributions on page request sequences that are not generated by any such Markov chain.

Instead of restricting $\mathcal{D}$, we assume only that the paging algorithm has access to some information about $\mathcal{D}$. The algorithm need only determine, for each pair of pages p and q, the probability that p will next be requested before q. We show that this information is enough to design a simple randomized algorithm that is 4-competitive against ON, and runs in time polynomial in the number of slots of fast memory. Even if the algorithm can only determine the probabilities approximately, it would be ensured competitiveness. Here ON is the best online algorithm on sequences drawn from $\mathcal{D}$.

A tournament is a complete graph in which each edge has a direction. It is a natural representation of a sports competition — the vertices of the graph are the players, while the direction of each edge indicates the winner of the individual matches. We generalize the notion of a tournament, by allowing edges to have real weights, rather than directions.

Definition 3.1 A *weighted tournament* is set V and a function $w : V \times V \to [0, 1]$, with the property that for each $u, v \in V$, $w(u, u) = 0$ and $w(u, v) + w(v, u) = 1$.

A weighted tournament represents the probability of outcomes of a future tournament, for example an edge with weight 3/4 from Edberg to Becker would indicate that with probability 3/4, Edberg will beat Becker. An ordinary tournament is just a weighted tournament in which all edge weights are 0 or 1.

Definition 3.2 A *dominating distribution* in a tournament $T = (V, w)$ is a probability distribution p on V such that for each vertex $v \in V$, if $u \in V$ is chosen according to p, then $E[w(v, u)] \leq \frac{1}{2}$.

Definition 3.3 The *dominating distribution algorithm* responds to each page fault by constructing the following weighted tournament: the vertices are the pages currently in fast memory. The weight $w(u, v)$ is the probability that u will next be requested no sooner than v. It finds a dominating distribution p for the tournament, and chooses a page to evict from the distribution p.

We first show that a dominating distribution always exists.

Theorem 3.4 *If $T = (V, w)$ is a weighted tournament, then T has a dominating distribution.*

Proof Consider the following linear program: minimize c subject to:

$$\sum_{u \in V} w(v, u) p(u) \leq c \qquad (\forall v \in V)$$

$$\sum_{u \in V} p(u) = 1$$

$$p(u) \geq 0 \qquad (\forall u \in V)$$

The claim is that the solution to this linear program is at most $\frac{1}{2}$. The dual linear program is to maximize d subject to:

$$\sum_{v \in V} w(v,u)q(v) \geq d \qquad (\forall u \in V)$$

$$\sum_{v \in V} q(v) = 1$$

$$q(v) \geq 0 \qquad (\forall v \in V)$$

It suffices to show that for any probability distribution q, there is a vertex u for which we have

$$\sum_{v \in V} w(v,u)q(v) \leq \frac{1}{2}.$$

To this end, consider the following derivation:

$$\begin{aligned}
&\sum_{u \in V} q(u) \sum_{v \in V} w(v,u)q(v) \\
&= \sum_{u,v \in V} w(v,u)q(u)q(v) \\
&= \sum_{u,v \in V:\, v<u} q(u)q(v)(w(v,u) + w(u,v)) \\
&= \sum_{u,v \in V:\, v<u} q(u)q(v)
\end{aligned}$$

Using the identity

$$1 = \left(\sum_{u \in V} q(u)\right)^2 = \sum_{u \in V} q(u)^2 + 2 \sum_{u,v \in V:\, v<u} q(u)q(v)$$

we have

$$\sum_{u \in V} q(u) \sum_{v \in V} w(v,u)q(v) = \frac{1}{2}\left(1 - \sum_{u \in V} q(u)^2\right) \leq \frac{1}{2}.$$

Thus there is some u for which $\sum_{v \in V} w(v,u)q(v) \leq \frac{1}{2}$, and the proof is complete. ∎

The proof of Theorem 3.4 shows that a dominating distribution can be found by solving a linear program, where the number of variables is just the number of page slots of fast memory.

Theorem 3.5 *The dominating distribution algorithm is 4-competitive against* On.

Proof When the dominating distribution algorithm has a page fault and must evict a page, let p be a random variable denoting the page that is evicted. The following property holds, by definition of the algorithm: for every page in q in fast memory, the probability that q is next requested no later than p is at least $1/2$. We can therefore use Lemma 2.5 to conclude that the dominating distribution algorithm is 4-competitive against On. ∎

4 The optimality of the median

Let $\mathcal{P}$ and $\mathcal{Q}$ be distributions over the reals. Let "$\mathcal{P} \ll \mathcal{Q}$" denote the event that independent random samples p and q (drawn from $\mathcal{P}$ and $\mathcal{Q}$) satisfy $p \leq q$. The key to the proof of Theorem 2.4 is that for any two distributions $\mathcal{P}$ and $\mathcal{Q}$,

$$\mathrm{med}(\mathcal{P}) \leq \mathrm{med}(\mathcal{Q}) \Rightarrow \Pr[\mathcal{P} \ll \mathcal{Q}] \geq 1/4. \tag{5}$$

If we could replace $1/4$ by some larger number we could prove a smaller competitive ratio for the median algorithm. We cannot increase $1/4$ in the above equation, however, since one can easily construct distributions $\mathcal{P}$ and $\mathcal{Q}$ so that $\mathrm{med}(\mathcal{P}) \leq \mathrm{med}(\mathcal{Q})$ but $\Pr[\mathcal{P} \ll \mathcal{Q}]$ is arbitrarily close to $1/4$.

We will show in this section that no matter what function we use in place of the median, we cannot increase the $1/4$ in (5). Intuitively, no function of distributions over the reals is better than the median at predicting the relative order of independent samples from the distributions.

We begin with some algebraic preliminaries. Define

$$g_n(x) = \begin{cases} 1 & n = 0 \\ 1 - x/g_{n-1}(x) & n > 0. \end{cases}$$

Fix a real x, and define $\alpha = \alpha(x) = \frac{1+\sqrt{1-4x}}{2}$ and $\beta = \beta(x) = \frac{1-\sqrt{1-4x}}{2}$. Notice that α and β may be complex, and that $t = \alpha$ and $t = \beta$ satisfy $t = x + t^2$. It follows that $t = \alpha$ and $t = \beta$ satisfy $t^3 = t^2 - tx = t^2 - (t^2 + x)x = (1-x)t^2 - x^2$. Hence, for all j

$$\begin{aligned} \alpha^{j+2} &= x\alpha^{j+1} + \alpha^{j+3} \\ \beta^{j+2} &= x\beta^{j+1} + \beta^{j+3} \\ \alpha^{j+4} &= (1-x)\alpha^{j+3} - x^2\alpha^{j+1} \\ \beta^{j+4} &= (1-x)\beta^{j+3} - x^2\beta^{j+1} \end{aligned} \tag{6}$$

Lemma 4.1 *If $\alpha^j \neq \beta^j$ for $1 \leq j \leq n+1$ then $g_n(x) - x = (\alpha^{n+3} - \beta^{n+3})/(\alpha^{n+1} - \beta^{n+1})$.*

Proof We use induction on n. For the $n = 0$ case we have

$$\begin{aligned} (\alpha^3 - \beta^3)/(\alpha^1 - \beta^1) &= \alpha^2 + \alpha\beta + \beta^2 \\ &= (\alpha + \beta)^2 - \alpha\beta \\ &= 1 - x \\ &= g_0(x) - x. \end{aligned}$$

Suppose the lemma is true when $n = j$, then

$$\begin{aligned} & g_{j+1}(x) - x \\ &= (1 - x/g_j(x)) - x \\ &= \frac{(1-x)g_j(x) - x}{g_j(x)} \\ &= \frac{(1-x)[(\alpha^{j+3} - \beta^{j+3})/(\alpha^{j+1} - \beta^{j+1}) + x] - x}{(\alpha^{j+3} - \beta^{j+3})/(\alpha^{j+1} - \beta^{j+1}) + x} \\ &= \frac{(1-x)(\alpha^{j+3} - \beta^{j+3}) - x^2(\alpha^{j+1} - \beta^{j+1})}{(\alpha^{j+3} - \beta^{j+3}) + x(\alpha^{j+1} - \beta^{j+1})} \\ &= \frac{\alpha^{j+4} - \beta^{j+4}}{\alpha^{j+2} - \beta^{j+2}}. \end{aligned}$$

The last step follows from (6). ∎

Let $\theta = \pi/(n+3)$ and $x_n = \left(1+\tan^2\theta\right)/4$. Then for $x = x_n$ we have

$$\begin{aligned}\alpha &= (1+i\tan\theta)/2 = e^{i\theta}/(2\cos\theta)\\ \beta &= e^{-i\theta}/(2\cos\theta).\end{aligned}$$

Hence $\alpha^{n+3} = \beta^{n+3}$ but $\alpha^j \neq \beta^j$ for $1 \leq j \leq n+1$, and the above lemma shows that $g_n(x_n) = x_n$. Notice that $x_n \to 1/4$ as $n \to \infty$.

Theorem 4.2 *For any function $\mathcal{F}$ mapping distributions to reals, and any $\epsilon > 0$ there exist distributions $\mathcal{P}$ and $\mathcal{Q}$ such that $\mathcal{F}(\mathcal{P}) \leq \mathcal{F}(\mathcal{Q})$ and $\Pr[\mathcal{P} \ll \mathcal{Q}] \leq 1/4+\epsilon$.*

Proof Fix n such that $x_n \leq 1/4+\epsilon$ and let α and β correspond to $x = x_n$. Let $p_1 = x_n$, and for $2 \leq j \leq n+1$ define $p_j = 1 - g_j(p_1)$. From the above expressions for α and β it is clear that for $1 \leq j \leq n+2$, $\alpha^j - \beta^j$ is i times a positive real. Hence for $1 \leq j \leq n$, $g_j(p_1) = (\alpha^{j+3} - \beta^{j+3})/(\alpha^{j+1} - \beta^{j+1}) + p_1 > 0$, so that $p_j < 1$. A simple induction argument shows that $g_j(p_1) + g_{n+1-j}(p_1) = 1$, so that $p_j + p_{n+1-j} = 1$. Thus $0 < p_j < 1$ for $1 \leq j \leq n$. Also $p_{n+1} = 1$.

We now define a set of distributions, $\{\mathcal{D}_j\}_{1\leq j\leq n+1}$. For $1 \leq j \leq n+1$ distribution $\mathcal{D}_j$ has mass p_j at j and mass $1-p_j$ at $n+j+1$.

	1	2				$n+1$		$2n$
$\mathcal{D}_1$	p_1	0				$1-p_1$		
$\mathcal{D}_2$	0	p_2				0		
			$\ddots$				$\ddots$	
$\mathcal{D}_n$				p_n	0			$1-p_n$
$\mathcal{D}_{n+1}$				0	1			0

For $1 \leq j \leq n$, $\Pr[\mathcal{D}_{j+1} \ll \mathcal{D}_j] = p_{j+1}(1-p_j)$. Since $p_{j+1}(1-p_j) = (1-g_{j+1}(p_1))g_j(p_1) = p_1$, each of these probabilities is p_1. Also, $\Pr[\mathcal{D}_1 \ll \mathcal{D}_{n+1}] = p_1$. Therefore, there must exist a j such that $\mathcal{F}(\mathcal{D}_{j+1}) \ll \mathcal{F}(\mathcal{D}_j)$ and $\Pr[\mathcal{D}_{j+1} \ll \mathcal{D}_j] \leq 1/4+\epsilon$, where $\mathcal{D}_{n+2}$ denotes $\mathcal{D}_1$. ∎

5 Lower Bounds

For our lower bounds, we consider conversations whose packet arrivals are generated by a restricted class of finite state Markov chains, where one time step corresponds to one transition of the Markov chain. The Markov chain has two types of transitions: *packet*-transitions where a packet is sent and *no-packet*-transitions where no packet is sent. The restriction in the IP-paging model that each conversation has a fixed inter-arrival time distribution (assumption 2) is satisfied by restricting the Markov chain so that all packet-transitions go to the same state.

Given a set of conversations generated this way, ON may be found by solving a linear program of size $s_1 s_2 \cdots s_n n\binom{n}{k}$, where n is the number of conversations, k is the number of circuits and for $i = 1, 2, \ldots, n$, s_i is the number of states in the Markov chain that describes the ith conversation.

In this section we prove lower bounds on the competitiveness of several online strategies. The lower bounds use several types of conversations. The *regular* type with parameter N will always have its packets coming every N'th time step. For the *bursty* type

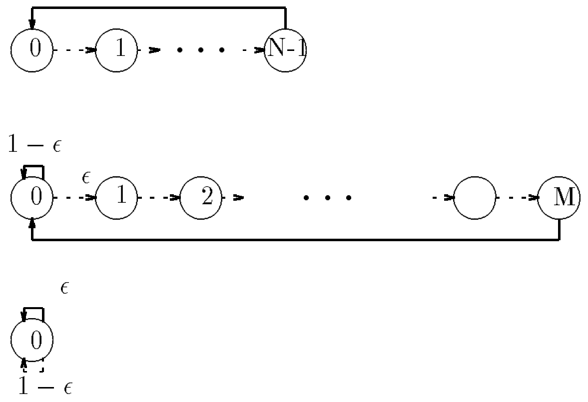

Figure 3: Markov chains describing regular type conversation (top), bursty type (middle) and geometric type (bottom). A solid edge denotes that a packet is sent. Each edge label is the probability the edge is taken, no label means probability is 1.

with parameters (ϵ, M), the next packet will arrive the next time step with probability $1-\epsilon$ and with probability ϵ it will arrive after M time steps. The *geometric* type with parameter ϵ will always have probability ϵ of having a packet in the next time step. These conversations can be described by finite state Markov chains. (See Figure 3).

We say a conversation of type bursty is "bursting" when the conversation is in state 0. By standard techniques we find that the stationary distribution for a bursty conversation C is $p_C(0) = 1/(1+\epsilon M)$ and $p_C(i) = \epsilon/(1+\epsilon M)$ for $i = 1, \ldots, M$, where $p_C(i)$ is the probability that C is in state i.

5.1 Non-constant Lower Bounds

This section gives lower bounds on the competitive ratio of LRU and two algorithms based on the optimal offline strategy. An optimal offline strategy is to close the circuit that is going to have a packet last. This suggests two online strategies: MLTBL, where the circuit most likely to be last is closed, and MET, which closes the circuit for which the expected time until the next packet is greatest. Unfortunately, neither of these algorithms has a constant competitive ratio against ON. This was also true in the Markov paging model, but the lower bounds presented here differ from and do not follow from those in [7].

Theorem 5.1 *The expected cost of MET divided by the expected cost of* ON *can be arbitrarily large, even in the case of 3 conversations and 2 circuits.*

Proof The first conversation is regular with parameter N. The second and third conversation are bursty with parameters (ϵ, M).

First consider the following algorithm A that always keeps a bursty conversation open when it is bursting and closes it otherwise. Note that there are 3 cases where A opens a new circuit:

1. Both 2 and 3 are bursting and 1 receives a packet. Here A opens two circuits: one for 1 and another for the bursty conversation that it just dropped in order to open 1.

2. A bursty conversation ends a burst (i.e., enters state 1). Here A may have to drop the bursty conversation and open 1 if it is not already open.

3. A bursty conversation begins (i.e. leaves state M). Here A has to open the bursty conversation.

Thus the probability that at a random time step that A will have to open a new circuit is bounded by $2p_1(N)p_2(0)p_3(0) + p_2(1) + p_3(1) + p_2(M) + p_3(M)$.

Next consider MET: if $M\epsilon > N$ it will always keep 1 open even when both 2 and 3 are bursting. Hence when both 2 and 3 are bursting then MET will open at least one circuit per time step. Thus its expected cost is at least $p_2(0)p_3(0)$.

Let $\epsilon = 1/N^2$ and let $M = N/\epsilon + 1 = N^3 + 1$. The expected cost for A is at most $2/N \cdot 1/(1+\epsilon M)^2 + 4\epsilon/(1+\epsilon M) = O(1/N^3)$, but the expected cost for MET is at least $1/(1+\epsilon M)^2 = 1/(N^2+2)$.

Thus the ratio of MET's expected cost to ON's expected cost is unbounded. ∎

Theorem 5.2 *In the IP-paging model MLTBL is no better than $\Omega(k/\log k)$-competitive.*

Proof Consider $k-1$ geometric conversations $G_1, G_2, \ldots, G_{k-1}$ with parameter $1/k$ and 2 bursty conversations B_1, B_2 with parameters $(5/k, 3k\log k)$. At any time each geometric conversations has at most $1/(k-1)$ chance of being last, since they all are equally likely to be last. However, when either B_1 or B_2 is bursting, their chance of being last is at least $2/k$. (With probability $5/k$ the burst ends and with probability $(1-(1-1/k)^{3k\log k})^{k-1} \leq 1 - 1/k$ all the geometric conversations will have a packet before the inter-burst gap ends.)

Thus when both B_1 and B_2 are bursting, MLTBL closes them alternately while holding all the geometric conversations open. Thus MLTBL will incur a cost of at least $p_{B_1}(0)p_{B_2}(0) = 1/(1+15\log k)^2$.

Consider the following online algorithm A that always keeps B_i open when it is bursting ($i = 1, 2$) and closed otherwise. When both B_1 and B_2 are bursting A closes G_1. The expected cost for A is at most $2p_{G_1}(R)p_{B_1}(0)p_{B_2}(0) + p_{B_1}(1) + p_{B_2}(1) + p_{B_1}(3k\log k) + p_{B_2}(3k\log k) = 2/k(1+15\log k)^2 + 20/k(1+15\log k) < 22/k(1+15\log k)$, where $p_{G_1}(R)$ is the probability that G_1 has a packet. The reasoning is similar to the that of the proof of Lemma 5.1.

Thus MLTBL is at most $\Omega(k/\log k)$ competitive. ∎

Using Lemma 2.3 we can show that MLTBL is $k+1$ competitive against ON. The circuit most likely to be last is last with probability at least $1/k$, hence the condition in Lemma 2.3 is satisfied with $c = k$. Thus we have fairly tight bounds for the competitiveness of MLTBL against the optimal online strategy.

The proof of the following is analogous to the proof in [7] that LRU does not have constant competitive ratio in the Markov paging model.

Theorem 5.3 *In the IP-paging model the competitive ratio of LRU against* ON *is exactly k.*

Proof Consider $k+1$ regular conversations $C_0, \ldots, C_k$ with parameter $k+1$, such that C_i receives a packet whenever the current time modulo $k+1$ is i. LRU opens a circuit at every time step, whereas the strategy that drop the most recently used conversation will only have to open a new circuit every k steps. Thus LRU can not be better than k competitive against the best online strategy. The result of Sleator and Tarjan [14] shows that LRU is k competitive against the best offline strategy even for arbitrary request sequences. ∎

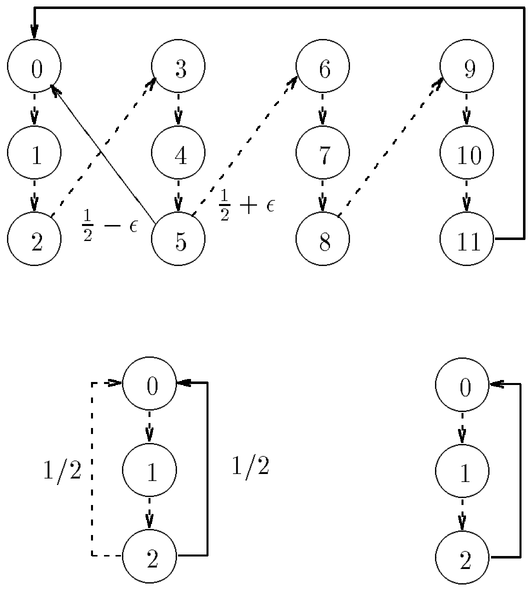

Figure 4: The conversations used in the lower bound for the median algorithm. The top Markov chain describes conversation 1, the bottom left describes conversation 2 and the bottom right describes conversation 3. A solid edge denotes that a packet is sent.

5.2 Lower bound for the median algorithm

In this section we present a 1.4 lower bound on the competitiveness of the median algorithm. This proof was found by extensive computerized searching. The best lower bound we have found is $1.511\cdots$, but the proof is too tedious to include here.

Consider the three conversations described in Figure 4, and let there be only 2 circuits. Conversation i is started in state $3-i$. Note that for $i = 1, 2, 3$, C_i only has a packet when the current time modulo 3 is i, so at each time step at most one conversation has a packet. Table 1 describes the stationary distribution for the median algorithm, when ϵ tends to 0. From this table it can be seen that the expected online cost tends to $7/36$ when ϵ tends to 0.

Now consider the algorithm A that always keeps the third conversation open. Consider the first conversation. It is always requested either at time 5 or at time 11. During the time from a packet on 1 up to and including the next packet on 1 there will be 2 opens if conversation 2 has a packet, otherwise there will be no opens. Thus we can count the expected number of opens as $2p_1(0)((1/2-\epsilon)p_2(\text{short}) + (1/2+\epsilon)p_2(\text{long}))$, where $p_2(\text{short})$ (respectively $p_2(\text{long})$) is the probability that 2 has a packet in a period of length 6 (respectively 12). It simple to see that $p_1(0) = 2/6(3-\epsilon)$, $p_2(\text{short}) = 1/2$ and $p_2(\text{long}) = 3/4$, thus the expected cost to A tends to $5/36$, when ϵ tends to 0. Thus we have the following lower bound:

Theorem 5.4 *In the IP-paging model the median algorithm is no better than* 1.4-*competitive against* ON.

old state		transition		new state		
time	open	req.	prob	time	open	cost
0, 2, 1	0, 1	1	1/27	1, 0, 2	0, 1	0
0, 2, 1	0, 1	–	1/27	1, 3, 2	0, 1	0
0, 5, 1	0, 2	–	1/27	1, 0, 2	0, 2	0
1, 0, 2	0, 1	2	1/27	2, 1, 0	1, 2	1
1, 0, 2	0, 2	2	1/27	2, 1, 0	0, 2	0
1, 3, 2	0, 1	2	1/27	2, 4, 0	1, 2	1
2, 1, 0	0, 2	–	1/27	3, 2, 1	0, 2	0
2, 1, 0	1, 2	–	1/27	3, 2, 1	1, 2	0
2, 4, 0	1, 2	–	1/27	3, 5, 1	1, 2	0
3, 2, 1	0, 2	1	1/54	4, 0, 2	1, 2	1
3, 2, 1	0, 2	–	1/54	4, 3, 2	0, 2	0
3, 2, 1	1, 2	1	1/54	4, 0, 2	1, 2	0
3, 2, 1	1, 2	–	1/54	4, 3, 2	1, 2	0
3, 5, 1	1, 2	–	1/27	4, 0, 2	1, 2	0
4, 0, 2	1, 2	2	2/27	5, 1, 0	1, 2	0
4, 3, 2	0, 2	2	1/54	5, 4, 0	0, 2	0
4, 3, 2	1, 2	2	1/54	5, 4, 0	1, 2	0
5, 1, 0	1, 2	0	1/27	0, 2, 1	0, 1	1
5, 1, 0	1, 2	–	1/27	6, 2, 1	1, 2	0
5, 4, 0	0, 2	0	1/108	0, 5, 1	0, 2	0
5, 4, 0	0, 2	–	1/108	6, 5, 1	0, 2	0
5, 4, 0	1, 2	0	1/108	0, 5, 1	0, 2	1
5, 4, 0	1, 2	–	1/108	6, 5, 1	1, 2	0
6, 2, 1	1, 2	1	1/54	7, 0, 2	1, 2	0
6, 2, 1	1, 2	–	1/54	7, 3, 2	1, 2	0
6, 5, 1	0, 2	–	1/108	7, 0, 2	0, 2	0
6, 5, 1	1, 2	–	1/108	7, 0, 2	1, 2	0
7, 0, 2	0, 2	2	1/108	8, 1, 0	0, 2	0
7, 0, 2	1, 2	2	1/36	8, 1, 0	1, 2	0
7, 3, 2	1, 2	2	1/54	8, 4, 0	1, 2	0
8, 1, 0	0, 2	–	1/108	9, 2, 1	0, 2	0
8, 1, 0	1, 2	–	1/36	9, 2, 1	1, 2	0
8, 4, 0	1, 2	–	1/54	9, 5, 1	1, 2	0
9, 2, 1	0, 2	1	1/216	10, 0, 2	1, 2	1
9, 2, 1	0, 2	–	1/216	10, 3, 2	0, 2	0
9, 2, 1	1, 2	1	1/72	10, 0, 2	1, 2	0
9, 2, 1	1, 2	–	1/72	10, 3, 2	1, 2	0
9, 5, 1	1, 2	–	1/54	10, 0, 2	1, 2	0
10, 0, 2	1, 2	2	1/27	11, 1, 0	1, 2	0
10, 3, 2	0, 2	2	1/216	11, 4, 0	0, 2	0
10, 3, 2	1, 2	2	1/72	11, 4, 0	1, 2	0
11, 1, 0	1, 2	0	1/27	0, 2, 1	0, 1	1
11, 4, 0	0, 2	0	1/216	0, 5, 1	0, 2	0
11, 4, 0	1, 2	0	1/72	0, 5, 1	0, 2	1

Table 1: The stationary distribution of the median algorithm. The table lists all the transitions in the Markov chain generated by the conversations and the median algorithm. The state of the Markov chain is the set of times since the last request for each conversation and which circuits are open.

References

[1] A. Borodin, S. Irani, P. Raghavan, and B. Schieber. Competitive paging with locality of reference. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 249–259, 1991.

[2] R. Caceres. The Pyramid IP to X.25 protocol interface: Merging DDN and PDN approaches. In *Proc. Uniform*, Washington, DC, 1987.

[3] C. Derman. *Finite State Markov Decision Processes*. Academic Press, New York, 1970.

[4] A. Fiat, R. Karp, M. Luby, L. McGeoch, D. D. Sleator, and N. Young. On competitive algorithms for paging problems. *Journal of Algorithms*, 12(4):685–699, 1991.

[5] P. A. Franaszek and T. J. Wagner. Some distribution-free aspects of paging performance. *Journal of the ACM*, 21:31–39, 1974.

[6] S. Irani, A. Karlin, and S. Phillips. Strongly competitive algorithms for paging with locality of reference. In *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, 1992.

[7] A. Karlin, S. Phillips, and P. Raghavan. Markov paging. In *Proceedings of the 33nd IEEE Symposium on Foundations of Computer Science*, pages 208–217, 1992.

[8] S. Keshav, C. Lund, S.J. Phillips, N. Reingold, and H. Saran. An empirical evaluation of virtual circuit holding time policies in ip-over-atm networks. *IEEE Journal on Selected Areas in Communications*, 13(8):1371–1382, 1995.

[9] P. A. W. Lewis and G. S. Shedler. Empirically derived models for sequences of page exceptions. *IBM J. Res. and Develop.*, 17:86–100, 1973.

[10] C. Lund, S. Phillips, and N. Reingold. Adaptive holding policies for IP over ATM networks. In *Proc. Infocom*, 1995.

[11] A. Rybczynski. X.25 interface and end-to-end virtual circuit service characteristics. *IEEE Trans. Comm.*, COM-28:500–509, 1980.

[12] H. Saran and S. Keshav. An empirical evaluation of virtual circuit holding times in IP-over-ATM networks. In *Proc. Infocom*, 1994.

[13] G. S. Shedler and C. Tung. Locality in page reference strings. *SIAM Journal on Computin*, 1:218–241, 1972.

[14] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, February 1985.