Optimal On-line Power Saving Strategies

Hossein Falaki
David R. Cheriton School of Computer Science
University of Waterloo
mhfalaki@cs.uwaterloo.ca

August 10, 2007

Abstract

We prove the worst case competitive ratio of on-line power-down strategies for systems with multiple power states to be $3 + 2\sqrt{2}$. For a system with k states, we present an $O(k^2 logk\ log(\frac{1}{\epsilon}))$ time algorithm that finds a near optimal on-line strategy.

1 Introduction

Recent advances in ubiquitous and mobile computing have made energy efficiency a critical concern. Dynamic Power Management (DPM) attempts to achieve energy efficiency by placing different components of a battery operated computer in "sleep state" when they are idle.

Many computer systems can reside in a high-cost and one or more low-cost states. They can only serve requests in the high-cost state, but residing in this state comes at a high per time unit energy cost. When there is no request to serve the system can transition to one of the low-power states to pay a lower per unit time cost, but at a fixed one-time transition cost. A dynamic power management scheme should balance the trade-offs between the two types of states to achieve minimum power consumption.

There are different approaches to DPM, including prediction strategies, stochastic modeling based strategies, and on-line strategies. Among these, the on-line strategies have certain advantages. It has been shown that they have better performance compared to other approaches [4]. In addition their worst case performance is theoretically bounded through competitive analysis.

In this paper we present the latest results regarding on-line dynamic power saving strategies. Next section briefly presents related work to dynamic power management. Section 3 formally defines the problem. We first present a simple on-line strategy and its competitive ratio for a specific version of the problem [3] in Section 4. In Section 5 we analyze the competitive ratio of the same strategy in the general setting and present an algorithm for finding a new optimal strategy. Finally in Section 7 we discuss some of the assumptions in recent literature [1], [4], [3] and possible future work.

2 Related Work

Prior to the introduction of on-line DPM algorithms, DPM solutions have been following either a predictive approach or a stochastic control approach [2]. The predictive approaches attempt to predict the length of the next idle period using past history of the system. Based on this prediction, the power management system chooses the optimal power state switching threshold. The prediction algorithm is either adaptive or non-adaptive. Non-adaptive predictive algorithms set the idleness threshold once and forever, whereas adaptive algorithms are affected by the observed input patterns. All these schemes make a single prediction and pay the overhead if the actual idle time is different from the predicted value.

Stochastic approaches make assumptions about the probability distribution of the job request patterns and formulate the DPM problem as an stochastic optimization problem. They are highly sensitive to the underlying assumptions, and proving any theoretical bounds on their performance is difficult.

Recently researchers have studied DPM as an on-line problem. Similar to any on-line algorithm, the power management system should decide about resource allocation before knowing all the input (i.e. the length of the idle period). On-line algorithms are analyzed based on their *competitive ratio*; the ratio of the on-line algorithm cost, to the cost of the optimal off-line algorithm.

The same general setting as the Dynamic Power Management probelm, although simpler, exists in many other areas of Computer Science. In a shared memory multiprocessing system, a process that is waiting for a locked resource must decide whether to spin or lock [5]. A gateway between a connection-oriented network and a connectionless network should decide when to drop a connection [6]. These are two-state examples of the same problem for which there is a 2-competitive deterministic on-line algorithm. When the input follows a known probability distribution the on-line algorithm can perform better and achieve $e/(e-1)$ expected competitive ratio.

In this paper we consider the general case, in which a system has multiple power sates. We present a 2-competitive strategy for systems with additive transition costs. We prove the same strategy to be $(3+2\sqrt{2})$-competitive in systems with arbitrary transition costs, but this is not necessarily the optimal strategy. Assuming the optimal strategy to be ρ-competitive, we present an algorithm that finds a $(\rho+\epsilon)$-competitive strategy for any ϵ value.

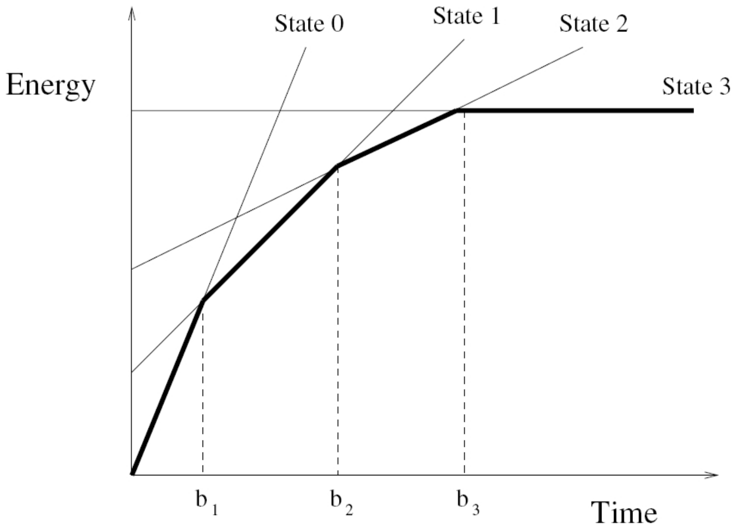

Figure 1: Energy consumption of different states and the optimal deterministic on-line strategy (in bold).

3 System Definition

For a formal system modeling lets assume a sequence of states, $< s_0, ..., s_k >$, and a vector of power consumption rates, $< \kappa_0, ..., \kappa_k >$, such that for $0 \leq i < j \leq k$, $\kappa_i > \kappa_j$. Also let $d_{i,j}$ be the transitioning cost from s_i to s_j. Any system can be modelled by a pair (K, d). A *strategy* or *schedule* is defined as $A = (S_A, T_A)$ where S_A is a subsequence of S and T_A is a sequence of transition times. We always require that $S(0) = s_0$ and $T(0) = 0$. $A(t)$ is the cost of the schedule for an idle period of length t. An *algorithm* gets a system as input and produces the optimal power-down strategy.

We can assume, without loss of generality, that power-up transition costs are zero. For any system (K, d) we define a new system (K, d') such that for every $i < j$, $d'_{i,j} = d_{i,j} + d_{j,0} - d_{i,0}$, and $d'_{j,i} = 0$. Any schedule in (K, d) has the same cost in (K, d'). Note that for all $i < j$, $d'_{i,j} < d'_{0,j}$, because we can safely assume $d_{i,j} < d_{i,0} + d_{0,j}$, otherwise to move from s_i to s_j the schedule can go to s_0 and then to s_j and pay less cost. For notation simplicity we use D_i instead of $d_{0,i}$ from now on.

4 Systems with Additive Costs

In an early work [3], Irani et. al. study the optimal power down strategy in the special case where the costs are additive (i.e. $d_{i,j} + d_{j,k} = d_{i,k}$ for all $i < j < k$), or when for all i, j, $d_{i,j} = 0$. The analysis of this strategy provides good intuition into the probelm in the general case.

In Figure 1 each power state, s_i, is represented with line of slope κ_i and a y-intercept of D_i. Each $\kappa_i t + D_i$ can be interpreted as the cost of the schedule that stays in state s_i for ever. The optimal *off-line* algorithm would always choose $min_i\{\kappa_i t + D_i\}$ (i.e. the lower envelope of Figure 1), therefore $OPT(t) = min\{\kappa_i t + D_i\}$.

The best known deterministic on-line strategy, called Lower Envelope Algorithm (LEA) [1], follows the optimal lower envelope curve at any time. That is, for all lines appearing in the lower envelope, if t_i denotes the first time when s_i becomes the optimal state (i.e. $\kappa_{i-1}t_i + D_{i-1} = \kappa_i t_i + D_i$, and $t_0 = 0$), LEA will transit to s_i at time t_i.

4.1 Analysis

For any $i \in [t_i, t_{i+1}]$, The on-line cost is

$$LEA(t) = \sum_{j=0}^{i-1} (\kappa_j(t_{j+1} - t_j) + d_{j+1,j}) + \kappa_i(t - t_i)$$

The ratio of the on-line cost to the optimal cost ($\frac{LEA(t)}{OPT(t)}$) would be maximized at $t = t_i$, therefore in the worst case:

$$\begin{aligned}
\frac{LEA(t)}{OPT(t)} &= \frac{\sum_{j=0}^{i-1}(\kappa_j(t_{j+1} - t_j) + d_{j,j+1})}{\kappa_i t_i + D_i} \\
&= \frac{\sum_{j=0}^{i-2} \kappa_j(t_{j+1} - t_j) + D_i}{\kappa_i t_i + D_i} \\
&= 1 + \frac{\sum_{j=0}^{i-1} \kappa_j(t_{j+1} - t_j) - \kappa_i t_i}{\kappa_i t_i + D_i}
\end{aligned}$$

We show that

$$\sum_{j=0}^{i-1} \kappa_j(t_{j+1} - t_j) - \kappa_i t_i \leq \kappa_i t_i + D_i$$

For every t_i $(\kappa_{i-1} - \kappa_i)t_i = D_i - D_{i-1}$, therefore:

$$\begin{aligned}
\sum_{j=0}^{i-1} \kappa_j(t_{j+1} - t_j) - \kappa_i t_i &= (D_1 - D_0) + (D_2 - D_1) + \\
&+ \ldots + (D_i - D_{i-1}) - \kappa_0 t_0 \\
&= D_i - D_0 - \kappa_0 t_0 \\
&\leq \kappa_i t_i + D_i
\end{aligned}$$

The last inequality holds, because $D_0 = 0$, $t_0 = 0$ and $\kappa_i, t_i \geq 0$. Therefore the competitive ratio of LEA is 2.

[1] Although according to our naming convention it is a strategy or schedule

5 The General Case

Generally, systems do not have additive costs, and the costs of transitions between states are not zero. In this section we find the competitive ratio of the LEA on-line schedule in the general setting. Next, an algorithm is presented that finds a near optimal schedule for any given system.

5.1 Worst Case Competitive Ratio

We will prove that the worst case competitive ratio of the LEA schedule is $3+3\sqrt{2}$. We consider two cases. The first case is when for all i, there exists some $\gamma > 1$ such that $D_i \leq \gamma D_{i-1}$. In the second case there is no such γ.

Algorithm 1 Finding S'

```
S' = {s_k}
j = k
for i = k − 1 downto 0 do
  if D_i ≤ D_j/γ then
    put D_i in S'
    j = i
  end if
end for
```

1. Assume $\gamma > 1$ exists, such that for all i, $D_i \leq \gamma D_{i-1}$. The for any $t \in [t_i, t_{i+1}]$, the ratio $\frac{LEA(t)}{OPT(t)}$ is maximized when $t = t_i$:

$$
\begin{aligned}
LEA(t) &= \sum_{j=0}^{i-1}(\kappa_j(t_{j+1}-t_j)+d_{j,j+1}) \\
&\leq \sum_{j=0}^{i-1}\kappa_j(t_{j+1}-t_j)+\sum_{j=1}^{i}D_j \\
&\leq OPT(t_i)+D_i\sum_{j=1}^{i}\gamma^{j-i} \\
&\leq OPT(t_i)+\frac{\gamma}{1-\gamma}D_i \\
&\leq (1+\frac{\gamma}{1-\gamma})OPT(t_i) \\
&= \frac{2\gamma-1}{\gamma-1}OPT(t_i)
\end{aligned}
$$

2. If no such $\gamma > 1$ exists, we consider a γ approximation of OPT, OPT', that uses a subset of S in which for all i, $D_i \leq \gamma D_{i-1}$. Let LEA' be

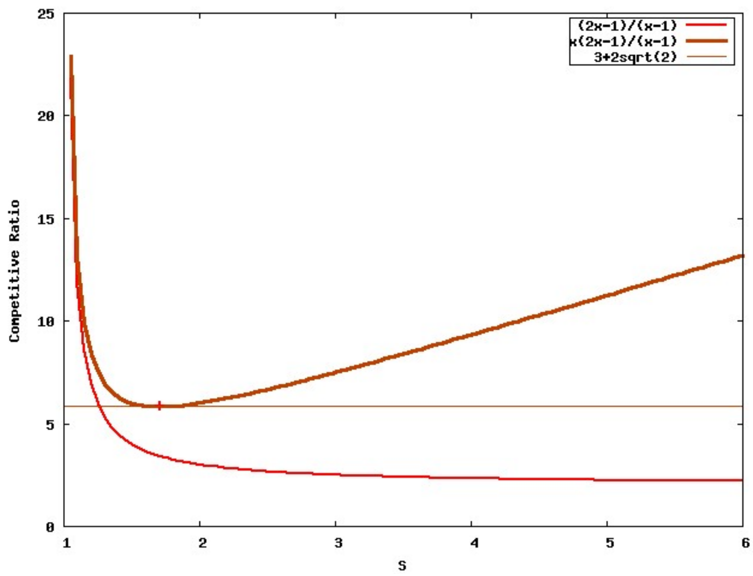

Figure 2: For $\gamma = 3 + 2\sqrt{2}$ the competitive ratio is $s + 2\sqrt{2} \approx 5.8284$.

the on-line schedule that uses the set of states of OPT'. Considering the analysis of the previous case we get $LEA'(t) \leq \frac{2\gamma-1}{\gamma-1} OPT'(t) \leq \frac{\gamma(2\gamma-1)}{\gamma-1} OPT(t)$

To get OPT', we start with $S' = \{s_k\}$. In every iteration we add a state according to Algorithm 1

We show that OPT' found by Algorithm 1 is indeed a γ-approximation of OPT. First note that $s_0 \in S'$, because $D_0 = 0$. OPT' will run the optimal off-line strategy using states in S'. Let $i, j \in S'$ and $l \notin S'$, s.t. $i < l < j$.

- If $t \in [t_i, t_{i+1})$, or $t \in [t_j, t_{j+1})$, then
  $OPT'(t) = OPT(t)$.
- If $t \in [t_l, t_{l+1})$, then
  $OPT'(t) = min(D_i + \kappa_i t, D_j + \kappa_j t)$ and $OPT(t) = D_l + \kappa_l t$.
  $D_l > D_i/\gamma, D_j/\gamma$ and $\kappa_l > \kappa_i, \kappa_j$, therefore $D_l + \kappa_l t > D_j/\gamma + \kappa_j t$ and $D_l + \kappa_l t > D_i/\gamma + \kappa_i t$ and hence $OPT(t) > OPT'(t)/\gamma$.

Figure 2 shows $\frac{2\gamma-1}{\gamma-1}$ and $\frac{\gamma(2\gamma-1)}{\gamma-1}$ for a range of $\gamma > 1$. Setting $\gamma = 1 + 1/\sqrt{2}$, gives the optimal competitive ratio $= 3 + 2\sqrt{2} \approx 5.8284$.

Theorem 1. *There is a* $(3 + 2\sqrt{2})$*-competitive strategy (schedule) for any system.*

6 Finding a Near Optimal Strategy

In this section we present an algorithm that will find a near optimal on-line power-down strategy. Theorem 1 states that in the worst case there always is a $(3 + 2\sqrt{2})$-competitive schedule, LEA, for any system. But it does not say anything about the lower bound. If the optimal on-line strategy is ρ-competitive, the deterministic algorithm by Augusttine et. al. [1] will find a $(p + \epsilon)$-competitive on-line strategy in time $O(k^2 logklog(\frac{1}{\epsilon}))$ where k is the number of states in the system.

We first prove that every ρ-competitive on-line strategy, has a representative "good" ρ-competitive strategy. We call these "good" strategies *eager*, and enumerate them to find if there is a ρ-competitive strategy. Unfortunately, the number of eager strategies exponentially depends on k. To further restrict the search space, we prove another restricting property for the representative strategies, called *earliness*. Eventually, a bisection search over values in $[1, 3+2\sqrt{2}]$, can give the $p+\epsilon$ on-line strategy. The following lemmas prove these properties for the ρ-competitive strategy.

Lamma 1. *If $A = (S, T)$ is a ρ-competitive strategy and s_l is the last state in S, then $\kappa_l \leq p.\kappa_k$.*

Proof. For A to be ρ-competitive, it should fall entirely below the $\rho.OPT$ convex in the energy-time diagram. If $\kappa_l > \rho.\kappa_k$, A will eventually cross the last line of $\rho.OPT$, which is a contradiction. □

Lamma 2. *For every strategy, $A(S, T)$, with finite competitive ratio the first time $\frac{A(t)}{OPT(t)}$ is maximized is a transition point in A.*

Proof. Assume t is the first time when $\frac{A(t)}{OPT(t)}$ is maximized, but is not a transition point, therefore $t \in (t_1, t_2)$ for two consecutive transition times in T, t_1 and t_2. If at t_1 A transitions to the last state in S, $t_2 = \infty$.

1. If t is not a transition point of OPT, there is ϵ s.t. $[t-\epsilon, t+\epsilon]$ does not include any transition point of OPT. We know $\frac{A(t-\epsilon)}{OPT(t-\epsilon)} < \frac{A(t)}{OPT(t)}$. Both A and OPT are linear in $[t-\epsilon, t+\epsilon]$, therefore $\frac{A(t)}{OPT(t)} < \frac{A(t+\epsilon)}{OPT(t+\epsilon)}$, which is a contradiction.

2. If t is a transition point of OPT, OPT starts a new line starting at time t, with lower slope than the line before t, therfore $\frac{A(t)}{OPT(t)} < \frac{A(t+\epsilon)}{OPT(t+\epsilon)}$, which is again a contradiction.

□

Definition 1. *For a strategy $A = (S, T)$ a transition at time t is ρ-eager if $A(t) = \rho.OPT(t)$. A is a ρ-eager strategy if if $A(t) = \rho.OPT(t)$ for all $t \in T$.*

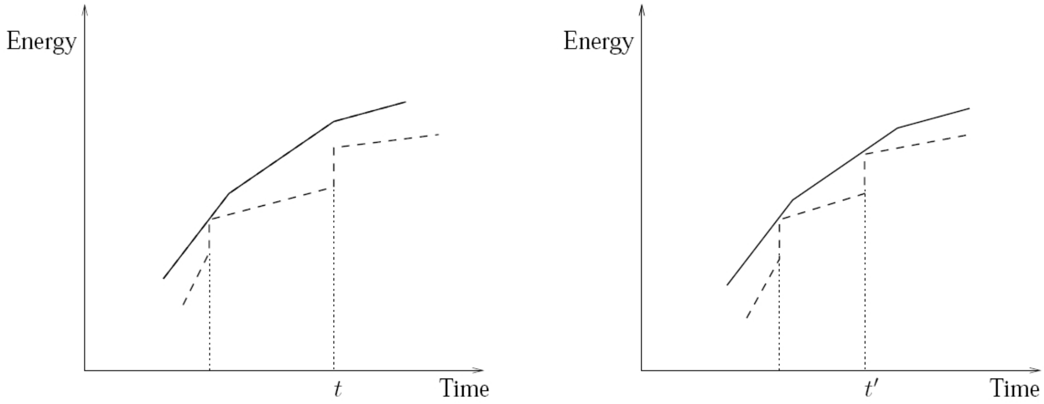

Figure 3: $\rho.OPT$ is represented with the solid line. t is the first non-eager transition time and t' is the eager transition time.

Lemmas 1 and 2 directly prove that if the last state of ρ-eager schedule, κ_s, is such that $\kappa_s \leq \rho.\kappa_k$, it is ρ-competitive.

Lamma 3. *For every ρ-competitive strategy, $A = (S, T)$, there exists an eager ρ-competitive strategy, $A' = (S, T')$.*

Proof. In Figure 3 the $\rho.OPT$ convex is plotted in solid line. Consider a non-eager strategy, A, (left) and its first non-eager transition, t, in dashed line. The transition point can be moved further to the left until it hits $\rho.OPT$ (right). The slope of the line before t in A is greater than the line after t, therefore the earlier transition, t', would only decrease the power consumption. The same process can be repeated for every transition point of A to get a ρ-eager strategy. □

Definition 2. *For each state, $s \in S$, $E_{s,\rho}$ is defined as the earliest time at which any on-line strategy can transition to s such that all its transitions are ρ-eager.*

Definition 3. *A transition to a state, s, is ρ-early if it happens at $E_{s,\rho}$. A strategy is ρ-early all its transitions are ρ-early.*

Lamma 4. *For every ρ-competitive strategy, $A = (S, T)$, there exists an early ρ-competitive strategy.*

Proof. Let s be the last state in S, and A' a strategy which has an early transition to s, as its last state. A' is ρ-competitive, because $\kappa_s \leq \rho.\kappa_k$. Note that by definition all transitions of A' are ρ-eager. We will prove that all transitions of A' are also ρ-early.

If A' is not ρ-early, there is at least one transition in A' that is not early. Let r be the last state with such a transition (i.e. A' transitions to r at time

t_1 s.t. $t_1 > E_{r,\rho}$). Therefore A' transitions to the state immediately after r, r', ρ-early (i.e. A' transitions to r' at time T s.t. $T = E_{r',\rho}$).

$E_{r,\rho}$ for any state, such as r, is well defined, therefore there is a strategy, such as B, that uses all the states in A' up to and including r and transitions to r at $E_{r,\rho}$. Lets make a strategy, A'', using all states and transition times of B followed by the states and transition times of A' starting at and including r'. For any $t \geq t_1$, $A''(t) = A'(t) - (A'(t_1) - A''(t_1))$. If $A''(t_1) < A'(t_1)$, then $A''(T) < A'(T)$ which means that we can shift the transition to r' in A'' to get a ρ-eager transition. This is contradicting with the our assumption that $T = E_{r',\rho}$.

Now we prove that $A''(t_1) < A'(t_1)$. Both transition points of A', at t_1, and A'', at t_2, are on the $\rho.OPT$ curve in the energy-time diagram. The slope of $\rho.OPT$ at time t_1 is greater than or equal to κ_r (of A') for all times $\geq t_1$. Therefore, $A'(t_1) = \rho.OPT(t_1) \geq \rho.OPT(t_2) + \kappa_r(t_1 - t_2) = A''(t_2) + \kappa_r(t_1 - t_2) = A''(t_1)$. In other words, $A'(t_1) \geq A''(t_1)$. □

Lemma 4 indicates that to check if a ρ-competitive strategy exists, we only need to consider all ρ-early strategies. Algorithm 2 uses dynamic programing to compute all $E_{i,\rho}$ for any given ρ value. If a ρ-early strategy exists the schedule will be printed. This algorithm is $O(K^2 logk)$ time.

Algorithm 2 Finding if a $(\rho + \epsilon)$-competitive strategy exists.

```
t[0] = 0
for i = 0 to k do
  t[i]: the earliest time a system can transition from any t[j]_{j=0...i} to s_i,
  ρ-eagerly
end for
if t[k] = inf then
  return NO
else
  print t[i]_{i=0...k}
  return YES
end if
```

Given any ϵ, Algorithm 2 can be used to perform a bisection search in the interval $[1, 3 + 2\sqrt{2}]$, and this will take $O(k^2 logklog(\frac{1}{\epsilon}))$ time.

7 Discussion and Future Work

In practice, service request times follow some probability distribution. Augusttine et. al. [1], consider the algorithm that produces the optimal strategy using the probability distribution of service request times. In favour of space, we skipped the probabilistic case.

No lower bound has been proved on the competitive ratio of on-line power-down strategies. Besides the lower bound open problem, there are two important directions for future work. Finding and analyzing the optimal on-line strategy in the face of uncertainty in system definition is one future direction. The power consumption specification of some computer systems may vary over time, and is not as fixed and well defined as assumed in [1]. The system definition should be extended to enable algorithmic consideration of such system uncertainties.

Some systems are capable of serving requests in states other than the active state, but at lower performance. Hence, the power saving strategy should consider both time and energy as performance metrics. Solving the problem in the more general framework where time is also considered, will be valuable.

References

[1] Augustine, J., Irani, S., and Swamy, C. Optimal power-down strategies. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 530–539.

[2] Benini, L., Bogliolo, A., and Micheli, G. A survey of design techniques for system-level dynamic power management, 2000.

[3] Irani, S., Shukla, S., and Gupta, R. Competitive analysis of dynamic power management strategies for systems with multiple power saving states. In *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings* (Paris, France, 2002), pp. 117–123.

[4] Irani, S., Shukla, S., and Gupta, R. Online strategies for dynamic power management in systems with multiple power-saving states. *Trans. on Embedded Computing Sys. 2*, 3 (2003), 325–346.

[5] Karlin, A. R., Li, K., Manasse, M. S., and Owicki, S. Empirical studies of competitve spinning for a shared-memory multiprocessor. In *SOSP '91: Proceedings of the thirteenth ACM symposium on Operating systems principles* (New York, NY, USA, 1991), ACM Press, pp. 41–55.

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