

Proceedings of the 1992 Conference on
Information Sciences and
Systems, March 18-20,
Princeton, NJ.

Optimal cell admission control in an ATM network node

Leandros Tassiulas, Yao Chung Hung, Shivendra S. Panwar

*Department of Electrical Engineering and
Center for Advanced Technology in Telecommunications
Polytechnic University*

Abstract

We consider here the problem of optimal cell admission control in an ATM network node. The objective is to minimize the average weighted cell rejection cost in a system with B buffer spaces, independent, identically distributed cell arrivals and deterministic service time. We formulate the problem as a Markov Decision Process and characterize the admission control policy that minimizes the objective. The optimal admission policy turns out to be of "multi-threshold" type.

1. Introduction

One of the main problems arising in the area of high speed communication networks is the design of control algorithms for the efficient sharing of the buffer space in an ATM node. Cells of different traffic types, as generated by a leaky bucket policing function or packetized video traffic, arrive at the node and are stored in a buffer until their transmission. When a cell finds the buffer full upon arrival, it is discarded from the system. The cell loss due to buffer overflow incurs a degradation in the overall system performance which is highly dependent on the class of the discarded cells. Certain traffic classes are more sensitive to cell losses than others. We can reduce the probability of discarding a cell of those classes due to buffer overflow, if we block the admission of cells belonging to classes whose loss has less impact on performance before the buffer is full. We study here how we can do this in an optimal manner. We formulate the problem as a Markov Decision Process and characterize the optimal policy using dynamic programming and sample path comparison arguments.

The space allocation problem in ATM networks has received considerable attention in the literature [8,1,2,3,4,5]. Lippman [8] proved the same result as we do here for an M/M/c queue, Kroner *et al* [1,2] studied this problem with a criterion of maximizing the offered load under two loss probability constraints for a two priorities system. Petr and Frost proposed an analysis framework for this problem[4] and found an efficient algorithm for determining the optimal set of thresholds to minimize the same criterion in a multi-priority system[5]. They also studied the problem of

minimizing the average weighted discarding cost per arrival in a system with a single buffer space[3], *i.e.* $B = 1$ as defined in next section. This paper proves the optimality, in terms of average weighted discarding cost per unit time, of the "multi-threshold" control policy in a system with buffer size $B \geq 1$.

2. The Model

The system is modeled by a single server queue. The queue has a buffer that can store at most B cells. This buffer will be called the main buffer in the following. Time is slotted and the transmission of a cell takes one slot. The cells are classified into L priority classes. The high priority classes are more sensitive to cell losses. Without loss of generality we assume that the priority of class l is higher than the priority of class $l+1$. The priority of a class is reflected by the cost that is incurred by the blocking of a cell of that class, as will become clear later. We assume that at most B_T cells of all priority classes may arrive in the system during one slot. This assumption is consistent with the structure of knockout-type ATM switches[7] or a switch with output queueing. All these cells are stored in a temporary buffer of length B_T . By the end of each slot a decision is taken about which cells will be admitted into the main buffer and where they are placed within it. The rest of the cells are discarded. We denote by $X_i^M(t)$ the class of the cell residing at the main buffer position i , $i = 1, \dots, B$ by the end of slot t ; $X_i^M(t) = 0$ if position i is empty at this time. We denote by $X_i^T(t)$ the class of the cell residing at position i of the temporary buffer $i = 1, \dots, B_T$; $X_i^T(t) = 0$ if this position is empty at this time. The vectors $\mathbf{X}^M(t) = (X_i^M(t) : i = 1, \dots, B)$, $\mathbf{X}^T(t) = (X_i^T(t) : i = 1, \dots, B_T)$, represent the main and temporary buffer occupancies at the end of slot t . Without loss of generality we may assume that in the temporary buffer, the cells are stored in decreasing priority order and in contiguous buffer spaces; that is, for $X_i^T(t) > 0$, $i > 1$, we have $0 < X_{i-1}(t) \leq X_i(t)$. The temporary buffer at the end of slot t contains cells that arrived during slot t only. We assume independent, identically distributed arrivals from slot to slot. The vector $\mathbf{X}(t) = (\mathbf{X}^M(t), \mathbf{X}^T(t))$ is a natural state variable and we use the notation $\{\mathbf{X}(t), t \geq 0\}$ for the stochastic process that describes the evolution of the system. The state space of that process is $\mathcal{X} = \mathcal{X}^M \times \mathcal{X}^T$ where $\mathcal{X}^M = \{0, 1, \dots, L\}^B$ and

This work was supported by the Center for Advanced Technology in Telecommunications, Polytechnic University and by the NSF under grant NCR-8909719.

$\mathcal{X}^T = \{0, 1, \dots, L\}^{B_T}$ are the spaces where the vectors $\mathbf{X}^M(t)$ and $\mathbf{X}^T(t)$ lie respectively. All the cells in the temporary buffer, by the end of each slot t , are either admitted and placed in the main buffer or rejected. We control the admission of cells in the main buffer. The control actions taken by the end of slot t are represented by the *admission* variables $A_i(t) \in \{0, 1, \dots, B\}$, $i = 1, \dots, B_T$ as follows. We have $A_i(t) = 0$ if either position i of the temporary buffer is empty or the cell stored in that position is blocked from admission into the system; we have $A_i(t) = j$ if the cell residing in position i of the temporary buffer is stored in position j of the buffer. The vector $\mathbf{A}(t) = (A_i(t) : i = 1, \dots, B_T)$ is called the admission vector at time t . Let $\mathcal{A} = \{0, \dots, B\}^{B_T}$ be the space where it lies; this is called action space in the following. We consider only cell admission control policies here and assume that the cells of the temporary buffer which are admitted in the main buffer are placed in consecutive positions at the end of the existing queue. Let $S_D(\mathbf{x})$ be the set of all admission vectors which satisfy the above assumption when the system is in state \mathbf{x} . At each slot t exactly one cell is transmitted. The cells in the main buffer are served in a FIFO manner. Given the state of the system at the end of slot t and the admission vector at that time, the main buffer occupancy vector by the end of slot $t+1$ is specified deterministically. Let $D : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{X}^M$ be a function such that $\mathbf{X}^M(t+1) = D(\mathbf{X}(t), \mathbf{A}(t))$. The state of the temporary buffer at the end of slot $t+1$ is determined completely from the arrivals during that slot. Under the assumption of i.i.d. arrivals the evolution of the system is Markovian. Given the state of the system at time t and the admission vector at that time, the probability distribution of the state at $t+1$ is completely determined by the function D and the probability distribution induced by the arrivals on \mathcal{X}^T . Let $p_{\mathbf{y}}$ be the probability that the temporary buffer has the configuration \mathbf{y} , $\mathbf{y} \in \mathcal{X}^T$ at the end of slot $t+1$. The transition probability $P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}) = Pr(\mathbf{X}(t+1) = \mathbf{x}' | \mathbf{X}(t) = \mathbf{x}, \mathbf{A}(t) = \mathbf{a})$ is given by

$$P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}) = \begin{cases} 0, & \text{if } \mathbf{x}'^M \neq D(\mathbf{x}, \mathbf{a}) \\ p_{\mathbf{y}}, & \text{if } \mathbf{x}'^M = D(\mathbf{x}, \mathbf{a}), \mathbf{x}'^T = \mathbf{y}, \mathbf{y} \in \mathcal{X}^T \end{cases} \quad (1)$$

An *admission policy* is any rule for selecting the admission variables at every time $t \geq 0$. This decision is made on the basis of the past system states $\{\mathbf{X}(s), t \geq s \geq 0\}$ and past decisions. We denote by G_D the set of policies.

When a cell of class l is dropped from the system then a cost c_l is incurred. We assume that the classes are indexed in decreasing priority, that is $c_l > c_{l+1}$, $l = 1, \dots, L-1$. By convention we set $c_0 = 0$. The total cost incurred when the system is in state \mathbf{x} and the admission actions that correspond to vector $\mathbf{a} \in S_D(\mathbf{x})$

are taken is

$$c(\mathbf{x}, \mathbf{a}) = \sum_{i=1}^{B_T} 1\{a_i = 0\} c_{x_i^T}, \quad \mathbf{x} \in \mathcal{X}, \mathbf{a} \in S_D(\mathbf{x}) \quad (2)$$

The sum in the right hand side of (2) accounts for the cost due to cell discarding. The blocking cost incurred at time t is denoted as $C(t)$ and is equal to $c(\mathbf{X}(t), \mathbf{A}(t))$. Our objective is to minimize the average blocking cost. The long run average cost associated with a policy $g \in G_D$ is defined by

$$J_g(\mathbf{x}) \stackrel{\text{def}}{=} \limsup_{T \rightarrow \infty} E_{\mathbf{x}}^g \frac{1}{T} \left[\sum_{t=0}^{T-1} C(t) \right], \quad \mathbf{x} \in \mathcal{X} \quad (3)$$

where $E_{\mathbf{y}}^g[\cdot]$ denotes the expectation with respect to the probability measure induced by the policy g on the state process starting in state \mathbf{y} . An admission policy g_D is said to be average cost optimal discarding policy if it minimizes (3) within G_D , i.e., if

$$J_{g_D}(\mathbf{x}) \leq J_g(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}$$

for any other policy $g \in G_D$. Under our assumptions about the arrival statistics, the optimization problem associated with (3) falls within the family of discrete time Markov Decision Processes (MDP's). Since the state space is finite, it is well known that an optimal policy exists and it can be taken in the class of Markov stationary policies[9]. A stationary policy g is identified by the functions $g_i : \mathcal{X} \rightarrow \{0, 1, \dots, B\}$, $i = 1, \dots, B_T$. When the system is controlled under policy g , at every time t we have $A_i(t) = g_i(\mathbf{X}(t))$ and in vector form $\mathbf{A}(t) = g(\mathbf{X}(t))$. In order to study the optimization problem associated with the long run average cost (3) we need to consider first the optimization problem associated with the β -discounted cost defined next.

The β -discounted cost ($0 < \beta < 1$) associated with a policy $g \in G_D$ is defined by

$$V_g^\beta(\mathbf{x}) \stackrel{\text{def}}{=} E_{\mathbf{x}}^g \left[\sum_{t=0}^{\infty} \beta^t C(t) \right], \quad \mathbf{x} \in \mathcal{X} \quad (4)$$

where $E_{\mathbf{x}}^g[\cdot]$ has the same meaning as in (3). An admission policy g_D^β is said to be β -optimal discarding policy if it minimizes (4) within G_D , i.e., if

$$V_{g_D^\beta}(\mathbf{x}) \leq V_g^\beta(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}$$

for any other policy $g \in G_D$. It is well known[9] that a β -optimal policy exists and it can be taken within the Markov stationary policies. The β -optimal cost associated with discarding policies is by definition

$$V_D^\beta(\mathbf{x}) = \inf_{g \in G_D} V_g^\beta(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}. \quad (5)$$

It is also well known that since the Markov decision process under consideration has finite state space, the β -optimal costs satisfy the dynamic programming equation

$$V_D^\beta(\mathbf{x}) = \min_{\mathbf{a} \in S_D(\mathbf{x})} \{c(\mathbf{x}, \mathbf{a}) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}) V_D^\beta(\mathbf{x}')\}. \quad (6)$$

In view of (1), equation (6) can be written as

$$V_D^\beta(\mathbf{x}) = \min_{\mathbf{a} \in S_D(\mathbf{x})} \{c(\mathbf{x}, \mathbf{a}) + \beta \sum_{\mathbf{y} \in \mathcal{X}^T} P_{\mathbf{y}} V_D^\beta(D(\mathbf{x}, \mathbf{a}), \mathbf{y})\}. \quad (7)$$

The necessary and sufficient condition for a stationary policy g to be a β -optimal discarding policy is

$$g(\mathbf{x}) = \arg \min_{\mathbf{a} \in S_D(\mathbf{x})} \{c(\mathbf{x}, \mathbf{a}) + \beta \sum_{\mathbf{y} \in \mathcal{X}^T} P_{\mathbf{y}} V_D^\beta(D(\mathbf{x}, \mathbf{a}), \mathbf{y})\}$$

3. Optimal cell discarding policy

In this section we study the problem of optimizing (5). We show that the optimal policy makes the admission decisions based only on the length of the main buffer and not on the class of packets in it. Furthermore, it is of threshold type with one threshold for each class of cells. We characterize the optimal policy for the β -discounted problem. Then by standard arguments[9] the same characterization is obtained for the average cost optimal policy.

3.1 State space reduction

In this section we show that under the optimal policy, the admission decisions at time t depend only on the number of cells in the main buffer and on the state of the temporary buffer. Based on that, we get a reduction of the state space. Let $l(\mathbf{x})$ be the number of cells in the main buffer when the system is in state \mathbf{x} . In the next lemma, we show that the β -optimal cost has the same value for any two states that correspond to the same temporary buffer state and the same number of cells in the main buffer.

Lemma 3.1: For any two states $\mathbf{x}_0 = (\mathbf{x}_0^M, \mathbf{x}_0^T)$, $\mathbf{x}_1 = (\mathbf{x}_1^M, \mathbf{x}_1^T)$ in \mathcal{X} such that $l(\mathbf{x}_0) = l(\mathbf{x}_1)$ and $\mathbf{x}_0^T = \mathbf{x}_1^T$, the value function for the β -discounted problem satisfies

$$V_D^\beta(\mathbf{x}_0) = V_D^\beta(\mathbf{x}_1).$$

Proof: For any policy g on \mathbf{x}_0 , since $\mathbf{x}_0^T = \mathbf{x}_1^T$, we can find a policy g' on \mathbf{x}_1 where g' discards a customer in \mathbf{x}_1^T if, and only if, g discards the same customer in \mathbf{x}_0^T .

Note that after these two decisions, the main buffer size remains equal for both states and the temporary buffer occupancy due to next arrival is the same, too. Thus g' can act as described above at the next decision instant.

Therefore, for any g , we have found a g' with the same value function.

This means $V_D^\beta(\mathbf{x}_0) \geq V_D^\beta(\mathbf{x}_1)$.

Similarly, we can prove $V_D^\beta(\mathbf{x}_1) \geq V_D^\beta(\mathbf{x}_0)$

$$\therefore V_D^\beta(\mathbf{x}_0) = V_D^\beta(\mathbf{x}_1).$$

Theorem 3.1: For any two states $\mathbf{x}_0 = (\mathbf{x}_0^M, \mathbf{x}_0^T)$, $\mathbf{x}_1 = (\mathbf{x}_1^M, \mathbf{x}_1^T)$ in \mathcal{X} such that $l(\mathbf{x}_0) = l(\mathbf{x}_1)$ and $\mathbf{x}_0^T = \mathbf{x}_1^T$, a β -optimal policy g_D^β satisfies

$$g_D^\beta(\mathbf{x}_0) = g_D^\beta(\mathbf{x}_1),$$

i.e. the β -optimal policies are identical.

Proof: Since $g_D^\beta(\mathbf{x}_1)$ incurs the same cost as $g_D^\beta(\mathbf{x}_0)$, the cost incurred by $g_D^\beta(\mathbf{x}_1)$ is $V_D^\beta(\mathbf{x}_0)$.

From Lemma 3.1, $V_D^\beta(\mathbf{x}_0) = V_D^\beta(\mathbf{x}_1)$, therefore, g_D^β acting on \mathbf{x}_1 achieves the optimal,

$$i.e. g_D^\beta(\mathbf{x}_0) = g_D^\beta(\mathbf{x}_1).$$

From the above results it is clear that when we consider cell discarding policies, we may consider the Markov decision process defined on the state space $\hat{\mathcal{X}} = \{0, 1, \dots, B\} \times \mathcal{X}^T$ which will be denoted by \mathcal{X} in the rest of this section. From now on the state of the process is $\mathbf{x} = (i, \mathbf{x}^T)$ where $i \in \{0, 1, \dots, B\}$ and $\mathbf{x}^T \in \mathcal{X}^T$. The action space associated with a state $\mathbf{x} = (i, \mathbf{x}^T)$ is equal to the common action space $S_D(\mathbf{x}')$ of all states $\mathbf{x}' \in \mathcal{X}^M \times \mathcal{X}^T$ such that $l(\mathbf{x}') = i$. An immediate implication of the above reduction of the state space is that the placement of the admitted cell in the main buffer is irrelevant as far as the optimal control problem is concerned. However, to maintain the FIFO ordering of cell transmissions, it may be convenient to consider them as being placed at the end of the queue. Therefore, the action vector should indicate for every cell in the temporary buffer whether it is admitted or not and can be taken to be binary. Indeed, the action vectors will be considered to be binary in the following. The transition operator $D(\cdot, \cdot)$ should indicate the number of buffers occupied in the main buffer for each action vector, therefore, in the new state space it can be considered to take values in Z^+ . If the action vector $\mathbf{a} \in S_D(\mathbf{x})$ has n nonzero elements, i.e. n cells are admitted, and $\mathbf{x} = (i, \mathbf{x}^T)$, then the transition operator is $D(\mathbf{x}, \mathbf{a}) = i + n - 1$ if $i \neq 0$, and $D(\mathbf{x}, \mathbf{a}) = i + n$ if $i = 0$.

3.2 Reduction of the action space

In this section we show that the reduction of the state space we achieved in the previous section implies a reduction in the action space from B_T -dimensional to one dimensional. More specifically we show that the optimal action is to accept the first n cells in the temporary buffer for some value of n . The placement

in the main buffer of the accepted cells is irrelevant. Consider the action vectors

$$\mathbf{a}_n = (a_j : a_j = 1, 1 \leq j \leq n, a_j = 0, n < j \leq B_T).$$

Theorem 3.2: The minimum in the right side of the dynamic programming equation (6) is achieved at one of the action vectors \mathbf{a}_n , $n = 0, \dots, B_T$ and the dynamic programming equation can be written as

$$V_D^\beta(\mathbf{x}) = \min_{n=0, \dots, B_T} \left\{ \sum_{j=n+1}^{B_T} C_{\mathbf{x}_j^T} + \beta \sum_{y \in \mathcal{X}^T} P_y V_D^\beta(1\{i > 0\}(i+n-1) + 1\{i = 0\}(i+n), y) \right\}. \quad (8)$$

The β -optimal discarding policy $g_D^\beta(\mathbf{x})$ can be considered to take values in $\{0, 1, \dots, B_T\}$ and it is

$$g_D^\beta(\mathbf{x}) = \arg \min_{n=0, \dots, B_T} \left\{ \sum_{j=n+1}^{B_T} C_{\mathbf{x}_j^T} + \beta \sum_{y \in \mathcal{X}^T} P_y V_D^\beta(1\{i > 0\}(i+n-1) + 1\{i = 0\}(i+n), y) \right\}. \quad (9)$$

Proof: Notice that for any action vector \mathbf{a}_n and state $\mathbf{x} = (i, \mathbf{x}^T)$, we have

$$\begin{aligned} C(\mathbf{x}, \mathbf{a}_n) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}_n) V_D^\beta(\mathbf{x}') \\ = \sum_{j=n+1}^{B_T} C_{\mathbf{x}_j^T} + \beta \sum_{y \in \mathcal{X}^T} P_y V_D^\beta(1\{i > 0\}(i+n-1) \\ + 1\{i = 0\}(i+n), y). \end{aligned} \quad (10)$$

in view of (10), in order to show (8), it is enough to show that

$$\begin{aligned} \min_{\mathbf{a} \in S_D(\mathbf{x})} \{C(\mathbf{x}, \mathbf{a}) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}) V_D^\beta(\mathbf{x}')\} \\ = \min_{n=0, \dots, B_T} \{C(\mathbf{x}, \mathbf{a}_n) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}_n) V_D^\beta(\mathbf{x}')\}. \end{aligned} \quad (11)$$

For any action vector \mathbf{a} with n nonzero elements we have

$$\begin{aligned} C(\mathbf{x}, \mathbf{a}) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}) V_D^\beta(\mathbf{x}') \\ \geq C(\mathbf{x}, \mathbf{a}_n) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}_n) V_D^\beta(\mathbf{x}'). \end{aligned} \quad (12)$$

This is so because first we have $D(\mathbf{x}, \mathbf{a}) = D(\mathbf{x}, \mathbf{a}_n) = 1\{i > 0\}(i+n-1) + 1\{i = 0\}(i+n)$, therefore

$$\beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}) V_D^\beta(\mathbf{x}') = \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}_n) V_D^\beta(\mathbf{x}'). \quad (13)$$

Also, as a result of arranging the temporary buffer in decreasing priority, we can easily see that

$$C(\mathbf{x}, \mathbf{a}) \geq C(\mathbf{x}, \mathbf{a}_n) \quad (14)$$

From (13) and (14) we have (12), and that implies

$$\begin{aligned} \min_{\mathbf{a} \in S_D(\mathbf{x})} \{C(\mathbf{x}, \mathbf{a}) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}) V_D^\beta(\mathbf{x}')\} \\ \geq \min_{n=0, \dots, B_T} \{C(\mathbf{x}, \mathbf{a}_n) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} P_{\mathbf{x}\mathbf{x}'}(\mathbf{a}_n) V_D^\beta(\mathbf{x}')\} \end{aligned}$$

so (11) follows.

3.3 The optimal policy

The characterization of the optimal policy is stated in *Theorem 3.3* after the next lemma.

Lemma 3.3: The value function associated with the β -discounted problem for discarding policies is convex in the sense that

$$\begin{aligned} V_D^\beta((i, \mathbf{x}^T)) - V_D^\beta((i-1, \mathbf{x}^T)) \\ \leq V_D^\beta((i+1, \mathbf{x}^T)) - V_D^\beta((i, \mathbf{x}^T)), \quad i = 1, \dots, B-1 \end{aligned}$$

Proof: Available upon request.

The following theorem characterizes the β -optimal policy.

Theorem 3.3: There exists a β -optimal discarding policy of the following form. There are thresholds $t_1 \geq t_2 \geq \dots \geq t_L$, where

$$\begin{aligned} t_j = \arg \min_{i \in \{1, \dots, B\}} \{C_j + \beta \sum_{y \in \mathcal{X}^T} P_y (V_D^\beta(i-2, y) \\ - V_D^\beta(i-1, y)) \geq 0\}, \end{aligned}$$

and such that a packet of class j in position k of the temporary buffer is accepted if and only if

$$t_j \geq i + k,$$

where i is the length of the main buffer. That is

$$u = \max\{k : i + k \leq t_{\mathbf{x}_k^T}\}.$$

Proof: Let us consider the case when $u > 0$ first. The control u minimizes the right side of the dynamic programming equation. Hence the difference of the right side of (8) evaluated at $n = u - 1$ and at $n = u$ is

$$C_{\mathbf{x}_u^T} + \beta \sum_{y \in \mathcal{X}^T} P_y (V_D^\beta(i+u-2, y) - V_D^\beta(i+u-1, y)) \geq 0$$

We show first that $i + u \leq t_{\mathbf{x}_u^T}$. If $i + u > t_{\mathbf{x}_u^T}$, then from *Lemma 3.3* and the definition of the thresholds we get

$$\begin{aligned}
0 &\leq C_{\mathbf{x}_i^T} + \beta \sum_{y \in \mathcal{X}^T} P_y(V_D^\beta(i+u-2, y) - V_D^\beta(i+u-1, y)) \\
&\leq C_{\mathbf{x}_i^T} + \beta \sum_{y \in \mathcal{X}^T} P_y(V_D^\beta(t_{\mathbf{x}_i^T} - 2, y) - V_D^\beta(t_{\mathbf{x}_i^T} - 1, y)).
\end{aligned}$$

The only condition for $t_{\mathbf{x}_i^T}$ to be the threshold is

$$\begin{aligned}
C_{\mathbf{x}_i^T} + \beta \sum_{y \in \mathcal{X}^T} P_y(V_D^\beta(i+u-2, y) - V_D^\beta(i+u-1, y)) \\
= C_{\mathbf{x}_i^T} + \beta \sum_{y \in \mathcal{X}^T} P_y(V_D^\beta(t_{\mathbf{x}_i^T} - 2, y) - V_D^\beta(t_{\mathbf{x}_i^T} - 1, y))
\end{aligned}$$

and $t_{\mathbf{x}_i^T} = i+u$. This is a contradiction.

Assume now that there is a $u' > u$ such that

$$i + u' \leq t_{\mathbf{x}_i^T} \quad (15)$$

Note that $C_{\mathbf{x}_i^T} + \beta \sum_{y \in \mathcal{X}^T} P_y(V_D^\beta(i+u'-2, y) - V_D^\beta(i+u'-1, y)) \geq 0$ from Lemma 3.3 and from the definition of threshold.

We will show that this is a contradiction as well.

Consider the terms

$$\begin{aligned}
C_{\mathbf{x}_i^T} + \beta \sum_{y \in \mathcal{X}^T} P_y(V_D^\beta(i+l-2, y) - V_D^\beta(i+l-1, y)) \\
\geq 0, \quad u \leq l \leq u' \quad (16)
\end{aligned}$$

The non-negativeness of the above terms is implied from Lemma 3.3 and the fact that the packets in the temporary buffer are stored in decreasing priority order.

The difference of the right side of (8) evaluated at $n = u$ and at $n = u'$ is

$$\begin{aligned}
\sum_{l=u+1}^{u'} C_{\mathbf{x}_i^T} + \beta \sum_{y \in \mathcal{X}^T} P_y(V_D^\beta(i+u-1, y) - V_D^\beta(i+u'-1, y)) \\
= \sum_{l=u+1}^{u'} (C_{\mathbf{x}_i^T} + \beta \sum_{y \in \mathcal{X}^T} P_y(V_D^\beta(i+l-2, y) \\
- V_D^\beta(i+l-1, y)))
\end{aligned}$$

From (16) it is clear that all the terms of the sum are positive, therefore u cannot be the action of the optimal policy. Note that for $u = 0$, we are interested only when $l(\mathbf{x}^T) > 0$. The proof is similar to the above except we begin with a difference of u and $u+1$ in the dynamic programming equation. Also note that when $l(\mathbf{x}^T) > 0$ and $i = 0$, $u = 0$ cannot be optimal.

This completes the proof.

3.4 Average cost optimal policy

We derive the corresponding results for the average cost case in this section. Specifically, we show that

the optimal policy is of threshold type, too. Before proving this we state a result from [9] that characterizes the average cost optimal policy of finite state space Markov Decision Processes.

Theorem 3.4: If the state space of an MDP is finite and there is some state, call it \mathbf{x}_0 , that is accessible from every other state regardless of which β -optimal policy is used, then there exists a bounded function $h(\mathbf{x}_i)$, $i \geq 0$, and a constant k such that

$$k + h(\mathbf{x}_i) = \min_{\mathbf{a}} [C(\mathbf{x}_i, \mathbf{a}) + \sum_{j=0}^{\infty} P_{\mathbf{x}_i, \mathbf{x}_j}(\mathbf{a}) h(\mathbf{x}_j)], \quad i \geq 0, \quad (17)$$

where $h(\mathbf{x}_i) = \lim_{n \rightarrow \infty} [V^{\beta_n}(\mathbf{x}_i) - V^{\beta_n}(\mathbf{x}_0)]$

for some sequence $\beta_n \rightarrow 1$.

Also there exists a stationary policy g^* such that

$$k = J_{g^*}(\mathbf{x}_i) = \min_g J_g(\mathbf{x}_i), \quad \text{for all } i \geq 0$$

and g^* is any policy that, for each state \mathbf{x}_i , prescribes an action that minimizes the right side of (17).

Due to the finite state space and the assumption about arrival statistics in our case, we can find a state to be the state \mathbf{x}_0 that is accessible from every other state.

Given the above theorem, in order to show the average cost optimal policy is of threshold type, we just need to show that $h(i, \mathbf{x}^T)$ is convex in i .

We do so in the following.

Theorem 3.5: The function $h(i, \mathbf{x}^T)$ is convex in i .

Proof: Let $h^{\beta_n}(i, \mathbf{x}^T) = V^{\beta_n}(i, \mathbf{x}^T) - V^{\beta_n}(\mathbf{x}_0)$. From Lemma 3.3,

$$\begin{aligned}
&h^{\beta_n}(i+1, \mathbf{x}^T) - h^{\beta_n}(i, \mathbf{x}^T) \\
&= (V^{\beta_n}(i+1, \mathbf{x}^T) - V^{\beta_n}(\mathbf{x}_0)) \\
&\quad - (V^{\beta_n}(i, \mathbf{x}^T) - V^{\beta_n}(\mathbf{x}_0)) \\
&= V^{\beta_n}(i+1, \mathbf{x}^T) - V^{\beta_n}(i, \mathbf{x}^T) \\
&\geq V^{\beta_n}(i, \mathbf{x}^T) - V^{\beta_n}(i-1, \mathbf{x}^T) \\
&= (V^{\beta_n}(i, \mathbf{x}^T) - V^{\beta_n}(\mathbf{x}_0)) \\
&\quad - (V^{\beta_n}(i-1, \mathbf{x}^T) - V^{\beta_n}(\mathbf{x}_0)) \\
&= h^{\beta_n}(i, \mathbf{x}^T) - h^{\beta_n}(i-1, \mathbf{x}^T)
\end{aligned}$$

By Theorem 3.4, $h(i, \mathbf{x}^T) = \lim_{n \rightarrow \infty} h^{\beta_n}(i, \mathbf{x}^T)$ for some sequence $\beta_n \rightarrow 1$, which implies that $h(i, \mathbf{x}^T)$ is convex in i .

Summary

We determined the optimal cell admission control in an ATM network node. We formulated the

problem as a Markov Decision Process and characterized the admission control policy that minimizes an average weighted cell rejection cost. The optimal admission policy turned out to be of "multi-threshold" type. Our results contribute to the design of efficient implementable protocols for buffer space management in ATM network nodes.

References

- [1] H. Kroner, "Comparative performance study of space priority mechanisms for ATM networks," Proceedings of the IEEE Infocom Conference, San Francisco, June 1990.
- [2] H. Kroner, G. Hebuterne, P. Boyer, A. Gravey, "Priority management in ATM switching nodes," IEEE Journal on Selected Areas in Communications, 418-427, April 1991.
- [3] D.W. Petr, V.S. Frost, "Optimal packet discarding: an ATM-oriented analysis model and initial results," Proceedings of the IEEE Infocom Conference, San Francisco, June 1990.
- [4] D.W. Petr, V.S. Frost, "Priority cell discarding for overload control in BISDN/ATM networks: an analysis framework," International Journal of Digital and Analog Communication Systems, Vol. 3, No. 2, April-June 1990.
- [5] D.W. Petr, V.S. Frost, "Nested threshold cell discarding for ATM overload control: optimization under cell loss constraints," Proceedings of the IEEE Infocom Conference, Florida, April 1991.
- [6] D.T. Luan, D.M. Lucantoni, "Throughput Analysis of an Adaptive Window-Based Flow Control Subject to Bandwidth Management," Proceedings of the 12th International Teletraffic Congress, Torino, Italy, 1988.
- [7] Y. Yeh, M.G. Hluchyj, A.S. Acampora, "The knockout switch: a simple, modular Architecture for high-performance packet switching," IEEE Journal on Selected Areas in Communications, 1274-1283, October 1991.
- [8] Steven A. Lippman, "Applying a new device in the optimization of exponential queuing systems," Operations Research, Vol. 23, No 4, July-August 1975.
- [9] S. Ross, *Introduction to dynamic programming*, Academic Press, New York, NY, 1983.