INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Université d'Ottawa · University of Ottawa

UNIVERSITY of OTTAWA

VBR TRAFFIC CONTROL in ATM NETWORKS

by

Tong Shan, B.Eng., M.Eng.

A Thesis

Submitted to the School of Graduate Studies and Research

In Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Information Technology and Engineering

University of Ottawa

Ottawa, Ontario, Canada

ABSTRACT

The VBR traffic control schemes are presented with the objectives of preventing network congestion, supporting end-to-end QoS on per-connection basis, and achieving a high level of resource utilization. First, an admission control algorithm is designed for the real-time VBR Traffic, which performs better than previous ones in terms of both the network bandwidth utilization and the computational time needed. Second, two lossless MLB traffic shapers are designed that can guarantee certain requirements for shaping buffer occupancy or delay bound. Third, the statistical traffic shaping scheme is put forward, and it can achieve a higher level of resource utilization than lossless shaping. Fourth, the impact of rate-controlling on end-to-end QoS is analyzed. Fifth, the guidelines and design criteria on controlling the VBR traffic are provided based on the research results in this thesis.

ACKNOWLEDGEMENT

First of all, I want to thank my supervisor Prof. Oliver W. W. Yang for his extremely helpful guidance and constructive suggestions throughout the process of this work. I really appreciate that Prof. Yang taught me many useful research approaches, and also gave me the opportunities to attend conferences and to work in the industry.

I am also grateful to the thesis review committee Prof. Jeremiah F. Hayes, Prof. Ahmed Karmouch, Prof. Shikharesh Majumdar, and Prof. Luis Orozco-Barbosa for their very helpful comments.

I want to express my appreciation to my colleagues in the computer communication research group for all the nice help.

I am grateful for the financial support from the MITEL corporation and from TRIO / CITO / MITEL internship.

Finally, I would like to thank my parents for their consistent encouragement and the deepest love. I appreciate that I have always had their understanding and support during the past four and half years.

TABLE OF CONTENTS

LIST of FIGURES

LIST OF TABLES

GLOSSARY

The following notations pertain to the remainder of this thesis. Enclosed in the bracket at the end of each definition is the section number where the notation first appears in this thesis.

$\propto_k$	the arrival time of the *k*th cell at the rate-controller, defined as the end of the slot over which the *k*th cell is completely received; (6.2.2)
$\hat{\alpha}_k$	the arrival time of the *k*th cell at the multiplexer; (6.3.1)
β	the maximum burst size of the regulated or shaped connection; (6.2.1)
β_j	the maximum burst size of the departure process from the connection *j*'s DLB shaper; (2.3)
$\beta_{j,n}$	the token buffer size of the *n*th leaky bucket of connection *j*'s MLB shaper; (2.3)
γ	the long-time average rate of the regulated or shaped connection; (6.2.1)
γ_j	the average rate of the departure process from the connection *j*'s DLB shaper; (2.3)
$\gamma_{j,n}$	the token rate of the *n*th leaky bucket of connection *j*'s MLB shaper; (2.3)
$\pi(j, i)$	the equilibrium probabilities for Markov Chain $\{D_k, S_k\}_{k\geq 1}$; (6.2.2)
$\pi_l(j, i, h)$	the equilibrium probabilities for Markov Chain $\{D_k, S_k, A_k\}_{k\geq 1}$; (6.2.3)
$\bar{\pi}$	the equilibrium probability vector for Markov process $\{D_k, S_k\}_{k\geq 1}$; (6.2.2)
$\hat{\pi}(j)$	the equilibrium probabilities for Markov Chain $\{\hat{D}_k\}_{k\geq 1}$; (6.3.1)
$\hat{\pi}_1(j, h)$	the equilibrium probabilities for Markov Chain $\{\hat{D}_k, \hat{A}_k\}_{k\geq 1}$; (6.3.2)
$\bar{\hat{\pi}}$	the equilibrium probability vector for Markov process $\{\hat{D}_k\}_{k\geq 1}$; (6.3.1)
θ_{ji}	the independent and random phase of the *i*th connection in traffic class *j*; (2.3)
ρ_j	the long-term-average-rate of the DLB regulated traffic flow *j* at the network entrance; (2.2.2)
$\tilde{\rho}_j$	the long-term-average-rate of the incoming traffic flow *j* at a node except at the first node (i.e., at the network entrance node); (2.2.2)
$\rho_{j,n}$	the slope of the *n*th line segment in the MLB constraint function I_j^*; (2.2.1)
$\tilde{\rho}_{j,n}$	the slope of the *n*th line segment in the MLB traffic constraint function $\tilde{I}_j^*$; (2.2.1)
σ_j	the burst size of the DLB regulated traffic flow *j* at the network entrance; (2.2.2)
$\tilde{\sigma}_j$	the burst size of the incoming traffic flow *j* at a node except at the first node; (2.2.2)
$\sigma_{j,n}$	the y-intercept of the *n*th line segment in the MLB traffic constraint function I_j^*; (2.2.1)

$\tilde{\sigma}_{j,n}$	the y-intercept of the nth line segment in the MLB traffic constraint function $\tilde{I}_j^*$; (2.2.1)
τ_j	the X-axis value at the flex point in the DLB regulated I_j^*, i.e., $\tau_j = \sigma_j / (P_j - \rho_j)$; (2.2.2)
$\tilde{\tau}_j$	the X-axis value at the flex point in the DLB traffic constraint function $\tilde{I}_j^*$, i.e., $\tilde{\tau}_j = \tilde{\sigma}_j / (\tilde{P}_j - \tilde{\rho}_j)$; (2.2.2)
$\tau_{j,n}$	the X-axis value at the nth flex point of the MLB traffic constraint function I_j^*; (2.2.2)
$\tilde{\tau}_{j,n}$	the X-axis value at the nth flex point of the MLB traffic constraint function $\tilde{I}_j^*$; (2.2.2)
τ_{Sj}	the maximum time interval length during which connection j's cells can be sent in the peak-rate P_{Sj} from the shaper; (2.3)
$\mu_{ji}(t)$	the utilized bandwidth at time t by the ith connection of class j; (2.4)
$\upsilon_{ji}(t)$	the multiplexing buffer requirement at time t from the ith connection of class j; (2.4)
ω_{R_j}	the activity factor of $R_{ji}(t)$; (5.2)
χ_j	the activity factor of $\mu_{ji}(t)$; (2.4)
Ψ_n	the probability distribution of the batch arrival size during one time slot; (2.2.4)
$A(L_S=0, L_M=L)$	the admissible region when $L_S=0$ and $L_M=L$; (5.3)
Γ	the number of slot-tokens that need to be removed when a cell departs from the data buffer in the proposed mechanism; (6.2.1)
Λ	the capacity of the slot-token pool in the proposed mechanism; (6.2.1)
$\Omega_{ji}(t)$	the on-off periodic departure process from the regulator for the ith connection of class j; (2.2.2)
Ψ_n	the size of batch arrival during one time slot; (2.2.4)
$a(n)$	the inter-arrival time distribution at a rate-controller; (2.2.3)
$\hat{a}(n)$	the inter-arrival time distribution at a multiplexer; (2.2.3)
$\hat{a}^{(m)}(n)$	the probability distribution of $\hat{A}_m$; (6.4)
$accud_j$	the cumulative scheduling delay of connection j computed at the destination; (3.3)
b_{Mj}	the maximum multiplexing buffer fill level from connection j's cells; (2.4)
b_{Sj}	the maximum shaping buffer fill level from connection j's cells; (2.3)
c_j	the bandwidth allocated to connection j; (4.3)
d_j	the end-to-end delay bound requirement of connection j; (3.1)
d_j^m	the scheduling delay of connection j at node m; (3.2)
$d^{(m)}(n)$	probability distribution of D_m, i.e., $d^{(m)}(n) = Pr\{D_m = n\}$; (6.4)
$\hat{d}^{(m)}(n)$	probability distribution of $\hat{D}_m$, i.e., $\hat{d}^{(m)}(n) = Pr\{\hat{D}_m = n\}$; (6.4)
e_{0j}	the bandwidth allocation of connection j; (2.4)

$e_{00,j}$, $e_{0Ls,j}$, $e_{0L,j}$	the effective bandwidth requirement of connection j when L_S=0, when 0<L_S<L, and when L_S=L, respectively; (5.3)
eted(n)	the probability distribution of end-to-end delay; (6.4)
i(t)	the number of arrivals during [t-1, t]; (A.2)
p	the probability that one cell arrives from a background stream during one slot; (2.2.4)
$P_{q,q'}^{(l)}$	the l-step transitional probability for Geo / D / 1 queue; (6.2.2)
$p(j, i; j', i')$	the transitional probability of the Markov chain $\{D_k, S_k\}_{k\geq l}$; (6.2.2)
$p_1(j,i,h;j',i',h')$	the transitional probability of Markov Chain $\{D_k, S_k, A_k\}_{k\geq 1}$; (6.2.3)
$\hat{p}(j;j')$	the transitional probability of the Markov chain $\{\hat{D}_k\}_{k\geq l}$; (6.3.1)
$\hat{p}_1(j,h;j',h')$	the transitional probability of the Markov chain $\{\hat{D}_k, \hat{A}_k\}_{k\geq l}$; (6.3.2)
$q_{ji}(t)$	the shaping buffer demand from the ith connection of class j at time t; (2.3)
r	the long-term average rate of the input stream at the rate-controller; (6.2.1)
sc(t)	the number of departures during [t-1, t]; (A.2)
t_d	the minimum X-axis value of the point in $SC_*(t)$ from which the horizontal distance to $I_j^*(t)$ is equal to D_j^m; (3.2.3)
t_k	the service epoch of the kth cell at the rate-controller, defined as the beginning of the slot over which this cell is transmitted; (6.2.1)
$\hat{t}_k$	the service epoch of the kth cell at the multiplexer; (6.3.1)
t_h	the X-axis value of the hth flex point in SC_*; (3.2.2)
u_f	the slope of $SC_*(t)$ at the point where $SC_*(t) = \rho_{j,1} \cdot \tau_{j,2}$; (3.2.3)
u_h	the slope of the hth segment in SC_*; (3.2.2)
$v_{j,l}$	the slope of the lth segment in S_j^*; (3.2.1)
$x_{j,l}$	the X-axis value of the lth flex point in S_j^*; (3.2.1)
$x^{(m)}(n)$	the probability distribution of X_m, i.e., $x^{(m)}(n) = \Pr\{X_m = n\}$; (6.4)
$\hat{x}^{(m)}(n)$	the probability distribution of $\hat{X}_m$, i.e., $\hat{x}^{(m)}(n) = \Pr\{\hat{X}_m = n\}$; (6.4)
A_j	the envelope function of the traffic shaper for connection j; (2.3)
A_j^{-1}	the inverse function of A_j; (3.2.3)
A_k	the inter-arrival time between the kth and (k+1)st cell at the rate-controller; (6.2.1)
$\hat{A}_k$	the inter-arrival time between the kth and (k+1)st cell at the multiplexer; (6.3.1)
A_{min}	the minimum possible value of the inter-arrival time at the rate-controller; (2.2.3)
$\hat{A}_{min}$	the minimum possible value of the inter-arrival time at the multiplexer; (6.3.2)
$A_{m,min}$	the minimum possible value of inter-arrival time at the rate-controller of the mth node; (6.4)

A_{max}	the maximum possible value of the inter-arrival time at the rate-controller; (2.2.3)
$\hat{A}_{max}$	the maximum possible value of inter-arrival time at multiplexer; (6.3.2)
A_m	the inter-arrival time process of the reference stream at the shaper of the *m*th node; (6.2.1)
$\hat{A}_m$	the inter-arrival time process of the reference stream at the multiplexer of the *m*th node; (6.3.2)
B_D	the data buffer capacity of the leaky bucket rate-controller; (6.2.1)
B_M	the multiplexing buffer capacity; (2.4)
B_S	the data buffer capacity at the traffic shaper; (5.2)
$B_{Scrh,j}$, $B_{Scrl,j}$	the critical values of the shaping buffer capacity for class j traffic, when statistical shaping is more beneficial than lossless shaping; (5.2)
C	output link bandwidth; (2.4)
$C_{C0,j}$, $C_{CLs,j}$	the critical bandwidth for class j traffic, when lossless shaping (L_S=0) and statistical shaping (L_S>0) are used, respectively; (5.3)
D_j	the end-to-end delay bound for connection *j*; (3.2.3)
D_j^m	the minimum upper bound of scheduling delay of connection *j* at the node *m* ; (3.1)
$\hat{D}_j^m$	the reassigned value of D_j^m ; (3.2.3)
$D(I_j^* \| A_j)$	the upper bound of the shaping delay of connection *j*; (3.1)
D_k	the cell delay incurred to the *k*th cell within the rate-controller; (6.2.1)
$\hat{D}_k$	the cell delay incurred to the *k*th cell at the multiplexer; (6.3.1)
D_m	delay process of the reference stream at the rate-controller of the *m*th node; (6.4)
$\hat{D}_m$	the queuing delay process of the reference stream at the multiplexer of the *m*th node; (6.4)
D_{max}	the maximum possible value of the waiting time at a rate-controller; (6.2)
$D_{on,j}$	the duration of the on period of the on-off periodic process $\upsilon_j(t)$ and $\mu_j(t)$; (2.4)
D_{Sj}	the upper bound of the shaping delay of connection *j*'s traffic; (4.2)
D_S	the delay bound at the rate-controller using the proposed leaky bucket mechanism; (6.2.1)
H	the number of the line segments in SC_* ; (3.2.2)
$H_{on,j}$	the duration of the on period of the on-off periodic process $R_j(t)$; (2.4)
I_j^*	the traffic constraint function of connection *j* at network entrance; (2.2.1)
$I_j[s, s+t]$	the number of cells arriving from connection *j* in time interval [s, s+t]; (2.2.1)
$\tilde{I}_j^*$	the traffic constraint function of the incoming flow *j* at a node except at the first node; (2.2.1)
K_j	the number of connections in class *j*; (2.3)
K_{S0}	the number of input connections at the traffic shaper; (5.2)

K_{Smax}	the maximum number of connections that can be accommodated at the shaper; (5.2)
$K(L_S>0, L_M=L-L_S)$	the maximum admissible number of connections when $L_S>0$ and $L_M=L-L_S$; (5.3)
L_j	the number of the line segments in S_j^*; (3.2.1)
L_C	the number of predefined delay classes; (1.1.2)
L	the QoS requirement of loss probability at the network node; (5.3)
L_M	the QoS requirement of loss probability at the traffic multiplexer; (5.3)
L_S	the QoS requirement of loss probability at the traffic shaper; (5.3)
M	the number of nodes on the path of the new or reference connection; (2.1)
$M_j(s)$	the moment generating function of $\mu_j(t)$; (5.3)
N	the number of existing connections at certain output link; (1.1.2)
N_b	the number of background traffic streams into a multiplexer; (2.2.4)
N_d	the number of homogeneous connections admitted when the end-to-end delay bound is *d*; (3.4)
N_f	the number of flex points on the spare capacity curve; (3.2.3)
N_j	the number of line segments in $I_j^*(t)$; (2.2.1)
$\tilde{N}_j$	the number of line segments in $\tilde{I}_j^*(t)$; (2.2.1)
N_{Sj}	the number of leaky buckets used in connection *j*'s MLB shaper; (2.3)
$\mathbf{P}$	the transitional probability matrix of Markov Chain $\{D_k, S_k\}_{k\geq 1}$; (6.2.1)
$\hat{\boldsymbol{P}}$	the transitional probability matrix of Markov Chain $\{\hat{D}_k\}_{k\geq 1}$; (6.3.1)
P_{LOSS}	the loss probability at the traffic shaper; (5.3.1)
P_S	the peak rate of shaped traffic from the rate-controller; (6.2.1)
P_j	the peak rate of the DLB regulated traffic flow *j* at the network entrance; (2.2.2)
$\tilde{P}_j$	the peak rate of the *j*th flow at a node except at the first node; (2.2.2)
P_{Sj}	the peak rate of the departure process from the connection j's shaper; (2.3)
P_{S0}, P_{SLs}, P_{SL}	the peak rate of the shaped stream when $L_S=0$, when $0<L_S<L$, and when $L_S=L$, respectively; (5.3)
$Q_b(t)$	the number of background cells in the multiplexing buffer at time *t*; (6.3.1)
Q_{Sj}	the upper bound of the shaping buffer occupancy for connection j; (4.2)
$R_{ji}(t)$	the extremal on-off periodic departure process from the traffic shaper of the *i*th connection of class *j*; (2.3)
S_k	the number of slot-tokens left behind by the *k*th cell's departure from the rate-controller; (6.2.1)
S_j^*	the service curve allocated to connection *j*; (3.2.2)
$\hat{S}_j^*$	the reallocated service curve of connection *j*; (3.2.3)
$S_j[s, s+t]$	the number of cells from connection j need to be served during the interval of [s, s+t] in order to support certain delay bound requirement; (3.2.2)
$SC_*(t)$	the spare capacity curve of the output link; (3.2.3)
$SC'_*(t)$	the updated spare capacity curve of the output link; (3.2.3)

$SC_{\bullet}^{v}$	the convex lower bound curve of $SC_{\bullet}(t)$; (3.2.3)
SC[s, s+t]	the number of cells that can be served by the spare capacity of an output link during the interval [s, s+t]; (3.2.2)
T	the minimum inter-departure time of the output traffic from the proposed mechanism; (6.2.1)
T_B	the maximum length of the busy period of the output link; (3.2)
T_j	the period of the on-off periodic process $\Omega_j(t)$; (2.2.2)
$T_{on,j}$	the duration of the on period of the on-off periodic process $\Omega_j(t)$; (2.2.2)
X_{k+1}	the inter-departure time from the rate-controller between the $(k+1)$st cell and its immediately previous departure cell; (6.2.3)
X_m	the inter-departure time process of the reference stream from the rate-controller at the mth node; (6.4)
$\hat{X}_m$	inter-departure time process of the reference stream from the mth multiplexer; (6.4)

The following acronyms pertain to the remainder of this thesis:

ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
CAC	Connection Admission Control
CBR	Constant Bit Rate
DLB	Dual Leaky Bucket
EDF	Earliest Deadline First
LB	Leaky Bucket
MLB	Multiple Leaky Bucket
QoS	Quality of Service
RCS	Rate-Controlled Service
SMG	Statistical Multiplexing Gain
UBR	Unspecified Bit Rate
VBR	Variable Bit Rate

Chapter 1

Introduction

High-speed networks must support communication services with heterogeneous traffic characteristics and diverse QoS (Quality of Service) requirements. Among the various traffic classes in the broadband networks, VBR (Variable Bit Rate) traffic poses a unique challenge in the network design. Unlike ABR (Available Bit Rate) or UBR (Unspecified Bit Rate) traffic which does not have stringent QoS requirements, VBR traffic class requires guaranteed QoS in terms of delay, delay jitter, and cell loss ratio, which requests some level of resource reservation. However, VBR traffic usually shows high burstiness. For instance, the video traffic usually shows a large fluctuation in rates due to scene changes and to the coding techniques used. The burstiness of VBR traffic makes it difficult to obtain high utilization of the network resources and to support the guaranteed QoS at the same time. The traffic pattern may also be distorted at the switches and routers, which makes the traffic flows even more bursty, thereby increasing its resource requirement along the path in order to guarantee the same QoS.

VBR traffic constitutes a significant part of the traffic in broadband networks, therefore VBR traffic control is critical in order to provide a desirable level of network performance. There are three major objectives of VBR traffic control: to prevent network congestion, to support the guaranteed QoS, and to achieve high network resource utilization.

1.1 Literature Review

We first review below several topics that are related to our research work.

1.1.1 Traffic control schemes

Many traffic control schemes can be found in the literature. These schemes have often been categorized into two categories: the closed-loop credit-based mechanism, and the open-loop rate-based mechanism [Onvu96, Stal94]. The closed-loop credit-based traffic control mechanism has been widely used in the traditional computer communication network protocols (e.g., X.25, TCP/IP), where the data communication applications constitute the major part of the traffic. The traditional data traffic (e.g., file transfer and email) is usually insensitive to the delay and delay variation, and the source sending rate can be easily slowed down in order to avoid the network congestion [Post81a, Post81b, Stal94].

In high speed networks, the propagation delay across the network becomes a significant part of the end-to-end delay, therefore, the traffic control scheme must be simple and able to operate at the speed of the communication link [ATMF96, FeVe90, Onvu96, SLCG93]. The traditional credit-based traffic control mechanism becomes unsuitable, since the feedback is usually outdated. In addition, the sending rate of a VBR traffic source (e.g., compressed video) is usually not controllable, i.e., it is usually hard for the network to slow down the sending rate of the VBR traffic source. Thus, the VBR flow control should be open-loop and rate-based rather than closed-loop and credit-based. The VBR source also requires guaranteed QoS, which requests some level of resource reservation. The above factors suggest that preventive congestion control instead of reactive congestion control should be used.

1.1.2 Admission control schemes

Admission control is an effective preventive congestion control scheme for VBR traffic; it prevents network congestion by controlling the number of active connections in the network.

Since the sending rates of VBR traffic sources are usually out of the control of network, the traffic characteristics and the QoS requirements of a new connection must be specified by the client at the connection setup time [ATMF96, Onvu96]. The admission control decides whether to admit the new connection or not based on: (i) the current usage of resources; (ii) the traffic characteristics and QoS requirements of the incoming traffic flow; (iii) the prediction of network performance. The new connection can be admitted only when its QoS requirements can be satisfied without degrading the QoS of the existing connections. Moreover, since the VBR traffic requires guaranteed QoS, some level of resource reservation is needed at every network element along the path of each VBR connection; hence resource reservation is one of the major tasks of the VBR admission control.

Recently, many admission control schemes have been proposed for VBR traffic, e.g., [FiKT97, Knig96, KnZh97, LiWF96, RaKZ96, WKZL96, ZhSh94]. The non-real-time VBR application requires the guarantee on cell loss ratio, and the real-time VBR traffic requires guarantees on delay, delay variation, and cell loss ratio. Because of the inherent burstiness and the stringent QoS requirements of the real-time VBR traffic, designing an admission control scheme for the real-time VBR traffic poses a uniquely challenging task.

Work has been done on the admission control for the real-time VBR traffic, in which a deterministic bound on the end-to-end delay is adopted as the measure of QoS [FeVe90, FiKT97, KnZh97, LiWF96, RaKZ96, WKZL96, ZhSh94]. Most of these works focused on the design of scheduling schemes to guarantee the delay bound, and the admission control is done through checking the schedulability conditions. There are two main issues addressed by these admission control schemes. First, the admission control must be fast enough to fit the real-time requirements. Second, it must be able to achieve efficient use of the network bandwidth.

However, none of these papers have investigated the two issues at the same time. For example, the work in [FeVe90, KnZh97, LiWF96, WKZL96] showed that reasonable bandwidth utilization can be achieved through checking the schedulability condition. They established that the complexity of their admission control is O(N), where N is the number of the existing connections at the time of the admission control; but they did no further work on reducing the computational complexity. The research in [FiKT97] provided an admission control algorithm for the earliest deadline first (EDF) scheduler and investigated the computational time in detail. A method was then proposed to reduce the computation complexity by classifying the delay requirements of connections, so that the complexity was reduced to O(Lc), where Lc is the number of the predefined delay classes. However, the traffic was simply characterized using the peak rate, the average rate and the maximum burst size, and results about the network utilization versus the delay bound requirement were not analyzed.

1.1.3 Traffic shaping schemes

After the connection establishment, it remains for the network to monitor and police the input traffic. The QoS requirements can be guaranteed only if the arriving traffic conforms to the clients' traffic specifications or is subject to traffic shaping / regulating at the network entrance. Traffic shaping usually reduces the traffic burstiness by spreading the transmission of a burst over time. By this way, traffic shaping protects the network resources from the misbehavior of sources, and prevents network congestion. In addition, the network resource utilization can also be improved by smoothing the traffic burstiness.

Various traffic shaping schemes have been presented in the literature. For example, a lossless smoothing algorithm was proposed in [LaCY96] where the short-term rate fluctuation is

smoothed through buffering. Multiple leaky bucket (MLB) shaper has been adopted as the generalization of traffic shaper by the Internet and ATM (Asynchronous Transfer Mode) standards, and has been discussed in [Cruz91a, Cruz91b, ElMi97, GGPS96, Graf97, ZhFe94]. It has been shown that the MLB shaper is not only simple to implement by passing the traffic stream through a series of leaky buckets, but it also minimizes both the shaping delay and the shaper buffer requirements [Cruz91a, Cruz91b]. The work in [Graf97] proposed traffic shaping at the end systems where the shapers at both the sender and the receiver were designed in order to accommodate the delay jitter. Lossless shaping at the ingress of the network was studied in [ElMi97]. It proposed to shape the less delay-sensitive traffic, so that the connection carrying capability can be increased. The papers [GGPS96, ZhFe94] discussed the rate-controlled service (RCS) discipline, in which the traffic of each connection is reshaped at every hop. The work in [GGPS96, Graf97] established that the resource requirements of a given connection can be minimized by shaping the traffic stream with a kind of minimum MLB shaper. The minimum MLB shaper means that, with certain requirement for the shaping buffer occupancy or shaping delay satisfied, the shaped traffic is with the minimum peak rate or the minimum burst size.

Summarizing the above studies, one can see that although the traffic distortion can be removed at the receiver, it increases the receiver buffer requirement and the end-to-end delay. Therefore, traffic shaping within the network (e.g., the RCS discipline) is a desirable way for providing the VBR traffic. It can improve the network performance in the following aspects: (i) The traffic can be made more predictable and less bursty; (ii) The QoS degradation can be alleviated; (iii) The requirements of expensive multiplexing buffer (usually more expensive than the shaping buffer) can be reduced; (iv) The link capacity requirements can be reduced. The RCS network model is to be detailed in section 2.1.

1.1.4 Resource utilization improvement schemes

Achieving high network resource utilization is an objective of VBR traffic control. Statistical multiplexing is an effective technique for improving the resource utilization, in which the statistical multiplexing gain is achieved at the price of occasionally discarding or excessively delaying a very small fraction of cells.

Statistical multiplexing has been widely studied in the past few years. Many researchers have worked on the resource allocation schemes at the multiplexer [ELMW95, GiHu91, GuAN91, Kell91, KeWC93]. However, their analyses are limited by the use of explicit probabilistic traffic models, because the stochastic properties of the traffic flows could be changed at a traffic multiplexer in an intractable way. The probabilistic traffic model is also very difficult to be policed at the network entrance or to be reshaped within the network. Besides, the network node model used in these papers only consists of a traffic multiplexer. Differently, as to be detailed in section 2.1, our network node model applying the RCS discipline has two components: a rate-controller and a multiplexer. Therefore, the results from the previous works are not directly applicable to our research work.

1.1.5 Analysis for the leaky bucket mechanism

The performance of the leaky bucket mechanism has been studied in many previous works [Berg91, Cruz91a, Cruz91b, ElMi91, ElMi97, GGPS96, Graf97, SLCG93,]. In [Cruz91a, Cruz91b, ElMi97, GGPS96, Graf97], the lossless service was guaranteed at the leaky bucket traffic shaper, and therefore the statistical performance was not analyzed. The statistical performance analysis of the leaky bucket scheme can be found in [Berg91, ElMi91, SLCG93], but their analyses are limited by the use of specific traffic models. For example, the paper

[SLCG93] made the assumption of Poisson arrivals which is not suitable to describe the traffic in high-speed networks. Markov modulated fluid model was used in [Berg91, ElMi91], where the input of the leaky bucket was essentially the multiplexed traffic of several sources. Thus it is not possible to investigate the individual connection's performance using their analytical methods. In addition, even with the restriction of traffic assumptions, analyses in the past works have never obtained a satisfying solution for the probability distribution of the cell delay and that of the inter-departure time. The paper [SLCG93] derived the Laplace transform of the waiting time and the inter-departure time, with the assumption of Poisson arrivals. The paper [ElMi91] presented the cell delay distribution with the use of Markov modulated traffic model, and only the mean and the second moment of the output rate were derived. The paper [Berg91] focused on analyzing the steady-state throughput and the blocking ratio, without discussing the distribution of cell delay and the inter-departure time process.

1.1.6 End-to-end QoS provisioning

The previous work on end-to-end QoS provisioning for individual VBR session can be found in [Cruz91b, FeVe90, GGPS96, GuAN91, KETB92, Kuro92, LaSt97, OhMM91, PaGa94, ZhFe94]. Typically, analyses are performed for a reference connection which traverses from its source to destination node. Deterministic bounds on end-to-end delay were derived in [Cruz91b, GGPS96, PaGa94, ZhFe94], where lossless service is guaranteed at the price of underutilization of network resources. [FeVe90, GuAN91, KETB92] derived the statistical bounds, but their analyses were limited by the specific traffic model assumptions. The probabilistic bounds were also derived in [Kuro92] without requiring any specific traffic assumption, however the result is not tight enough to be useful in practice. Approximate evaluation of the end-to-end QoS for

individual connection under the assumption of generally and independently distributed inter-arrival processes can be found in [LaSt97, OhMM91]. The paper [OhMM91] used the convolution method to derive the waiting time distribution and the inter-departure time distribution, which has high computational cost. The work in [LaSt97] proposed a peak-rate enforcement scheduling approach and analyzed the end-to-end performance of a tagged stream that is scheduled using the proposed approach. However, scheduling conflicts may become a problem when this scheduling approach is used for a number of streams.

The major difficulty of end-to-end QoS provisioning is that the characteristics of a traffic flow can be changed in an intractable way when it traverses several nodes. Motivated by this fact, the rate-controlled service (RCS) discipline was proposed [ZhFe93, ZhFe94, GGPS96]. The RCS discipline is not only an effective way to guarantee the end-to-end QoS, but it also simplifies the end-to-end performance analysis which can be extended easily from the single node analysis. However, only the deterministic bounds on the end-to-end QoS were derived in the previous works on RCS discipline, and the lossless service may result in under-utilization of resources.

1.1.7 Traffic models

Two categories of traffic models are usually used in characterizing the VBR traffic: stochastic traffic models and deterministic traffic models. Most analyses for VBR traffic are performed using stochastic models, such as the Markov Modulated processes [HeLu86], the generally and independently distributed inter-arrival processes [OhMM91], the self-similar models [GaWi94], and the (R_k^i, k) model [Kuro92]. The purpose of using the stochastic models is to try to capture the burstiness and correlations of the VBR traffic sources, and thereby to achieve the statistical

multiplexing gain. However, most stochastic models are not powerful enough to characterize the realistic VBR sources, and they are too complex for practical implementation of admission control and traffic policing. Furthermore, the stochastic property of the traffic flow can be changed in an intractable way when traffic traversing several nodes, which makes it hard to provide guaranteed end-to-end QoS in a conventional network model with general topology.

On the other hand, a number of deterministic traffic models have been proposed to characterize the worst case behavior of the VBR sources. For example, the (Xmin, Xave, I, Smax) model [FeVe90], the multiple-leaky-bucket model [Cruz91a, WKZL96], and the D-BIND model [KnZh97]. The deterministic traffic models can be easily verified and policed by the network. However, it is hard to achieve the statistical multiplexing gain when using the worst-case traffic model. Recently, in order to achieve the statistical multiplexing gain, the H-BIND model [Knig96] and a periodic on-off model with random phases [ElMW95, MiMo95] were proposed. In the proposal of the kind of periodic on-off process with random phases, the paper [MiMo95] combined the technique of the MLB traffic model with the large deviation theory. In the H-BIND model, the technique of the D-BIND model is combined with the aggregation approach.

In addition, the continuous fluid flow model has often been used in many previous papers to simplify the analysis [ElMW95, GGPS96, MiMo95, OhMM91]. When this model is used, the packetization time is not taken into account.

1.1.8 Network models

Analysis for an ATM network with general topology is usually performed by decomposing the network into nodes [OhMM91, Kuro92, KeTB92, RaKZ96, LiWF96, LeLD96, FiKT97, LaSt97,

MaSB97]. This decomposition approach allows a modular analysis of network. A network node can be a switch, a router, or other switching elements. The assumption of a non-blocking switching fabric is often used, which means buffering is only needed on the output ports. In addition, the fixed-length cell size in the ATM network leads to a discrete-time model, with the slot size equal to the transmission time of a cell by a full capacity link. Therefore, an ATM node can be modeled by a discrete-time single-server queue, where the queue and server correspond respectively to the output buffer and output link. Furthermore, any referenced ATM connection that traverses a sequence of nodes can be modeled by a cascade of queues.

In such a network environment, all the traffic flows that are routed to the same output link will be multiplexed at the same output buffer and served by the same server. The properties of an input traffic flow may be destroyed by the interaction among connections when it traversing several nodes. Thus, even if QoS guarantees can be made at a single node, it is hard to evaluate the end-to-end performance. In addition, admitting a new connection may perturb the traffic of exiting connections, necessitating a re-evaluation of the end-to-end QoS of these connections. The perturbation is not limited to the connections that share a server with the new connection, and may spread to other connections. This may lead to highly complicated admission control which is not suitable for the real-time applications.

Motivated by this fact, a network model using the rate-controlled service (RCS) discipline was proposed [ZhFe93, ZhFe94, GGPS96]. In this network model, the traffic pattern of each connection is reconstructed at every hop to ensure that the traffic offered to the multiplexer conforms to certain traffic characterization. By this way, the RCS discipline makes the characterization of the traffic flow more tractable, and therefore it is suitable for providing

end-to-end performance guarantees to individual connection in high speed networks. The network model using the RCS discipline will be detailed in section 2.1.

1.2 Objectives and Motivations of the Research Work

Generally, our research work on VBR traffic control has the following three major objectives: to prevent network congestion, to support the guaranteed QoS, and to achieve high network resource utilization.

Although a lot of work has been done on VBR traffic control as reviewed before, there is room for further improvement.

First, as reviewed in section 1.1.8, most of the conventional network models have deficiencies in supporting end-to-end QoS and in preventing network congestion. Therefore, we like to study the RCS (Rate-Controlled Service) network model that outperforms the conventional network model in several aspects: congestion control, resource utilization, and end-to-end QoS provisioning.

Second, as reviewed in section 1.1.2, there are two major issues involved in the admission control algorithm design for the real-time VBR traffic. However, none of the past works have investigated these two major issues at the same time.

Third, as reviewed in section 1.1.3, the resource requirement of a given traffic flow can be minimized by the minimum MLB shaper. But there appears to be no work on the minimum MLB shaper designs that can satisfy a specified shaping delay bound or a specified shaping buffer occupancy bound. Besides, there appears to be no study on the node delay bound performance of the minimum MLB shaper.

Fourth, as reviewed in section 1.1.3, only lossless shaping was discussed in the previous work, and the lossless service is provided at the cost of low resource utilization. However, there appears to be no work on improving the resource utilization by allowing a small loss probability at the shaper.

Fifth, as reviewed in section 1.1.6, only the deterministic bounds were derived in the past works regarding the impact of rate-controlling on the end-to-end QoS. The deterministic QoS is guaranteed at the expense of low resource utilization. When statistical multiplexing is used with the purpose of achieving higher resource utilization, the QoS must be analyzed in forms of probabilistic bounds.

So specifically, we want to achieve the following objectives:

(1) To design an efficient admission control for the real-time VBR traffic in order to achieve high utilization of network bandwidth and to fit the real-time requirements.

(2) To present two minimum multiple-leaky-bucket traffic shaper designs that can satisfy a specified shaping delay bound or a specified shaping buffer occupancy bound, and to analyze the node delay bound performance of the minimum MLB shaper.

(3) To study statistical traffic shaping, which allows a small loss probability at the shaper, with the purpose of improving the resource utilization.

(4) To analyze the impact of rate-controlling on the end-to-end performance in terms of the statistical bounds.

1.3 Methodologies and Approaches

Corresponding to the specific objectives listed in the above section, we present in this section the methodologies and approaches used in our research.

In our work on designing an admission control algorithm for the real-time VBR traffic, queuing analysis is carried out to analyze the node delay bound and the end-to-end delay bound for each VBR traffic flow. The admission control algorithm is then put forward based on these analysis results.

In the process of designing the lossless MLB traffic shaper, we also conduct the queuing analysis for the upper bound of shaping buffer occupancy, the shaping delay bound, and the node delay bound. Based on these analytical results, the minimum MLB shaper designs are presented.

In our work on statistical traffic shaping, large deviation theory is used to investigate the condition under which our proposed statistical shaping scheme is more beneficial than lossless shaping. As a branch of the probability theory, the large deviation theory is concerned with calculating the asymptotic rates with which certain probabilities converge to zero [Buck90]. Specifically, the Chernoff bound in the large deviation theory is used to estimate the cell loss probability at the shaper.

In our analysis for the end-to-end network performance, we use the embedded Markov Chain analysis approach to study the statistical performance of the rate-controller and traffic multiplexer. In order to verify the soundness of our analysis, we carry out performance evaluation.

With the goal of investigating the impact of rate-controlling on the end-to-end network performance, we conduct the simulation using the simulation tool OPNET [Mil97]. OPNET is an object-oriented simulation package, by which we can set up queuing models on three levels, i.e., the network, the node, and the process. The probe function in OPNET not only makes it easy to obtain the mean and variance for certain random variables, but it can also obtain probability distributions of delay, delay variation, and inter-departure time, etc. In our simulations, various

network, node, and process models are designed. First, a tandem network model is designed, in which the RCS discipline is applied at each node. Second, the node model is designed, and it consists of a rate-controller and a multiplexer. Third, two process models are designed. One is for implementing the leaky bucket mechanism, and the other is designed to implement FCFS multiplexing.

1.4 Contributions and Organization of Thesis

The contributions of this thesis are:

(1) An efficient bandwidth allocation scheme designed for the admission control of real-time VBR traffic.

Unlike previous studies, we make use of the service curve and the spare capacity curve. We show that our bandwidth allocation algorithm performs better than previous ones in terms of both the network bandwidth utilization and the computational time needed.

(2) Two minimum MLB shaper designs for lossless traffic shaping.

We present two minimum multiple leaky bucket (MLB) shaper designs that can guarantee the specified shaping delay bound or the specified worst-case shaping buffer occupancy. Furthermore, we show that the worst-case node delay of the minimum MLB shaper which guarantees the specified shaping delay bound is not longer than that when no traffic shaper is used.

(3) A statistical shaping scheme to improve resource utilization.

Unlike previous work, we allow a small loss probability at the traffic shaper. We show that statistical shaping can achieve a higher level of network resource utilization than lossless shaping when we apply certain design criteria in the network node design.

(4) The impact of rate-controlling on the end-to-end statistical performance.

Different from past work, we analyze the impact of rate-controlling on the end-to-end performances in terms of statistical bounds. First, we study the queuing performance of a rate-controller. A new operating mechanism is put forward for the leaky bucket rate-controller, and it has the advantage of allowing a simple queuing analysis. Then, the traffic distortion incurred at the multiplexer is studied. Finally, the end-to-end performance is obtained.

(5) Guidelines and design criteria on controlling VBR traffic.

On the basis of summarizing the research results in the above topics, we provide the guidelines and design criteria on controlling VBR traffic.

The rest of this thesis is organized as follows. Chapter 2 gives the network and traffic models, and describes the operations of the traffic shaper and multiplexer. Chapter 3 presents our admission control algorithm for the real-time VBR traffic. Chapter 4 gives the design of lossless MLB traffic shaper. In chapter 5, we put forward the statistical traffic shaping scheme. In chapter 6, we study the impact of rate-controlling on the end-to-end statistical performance. In chapter 7, we present the guidelines on controlling the VBR traffic. Finally, chapter 8 concludes the thesis.

Chapter 2

Models and Operations

Since the network and traffic models must be chosen before any analysis or design can be carried out, we present in this chapter the network and traffic models used in our research. We shall describe the operations of the traffic shaper and traffic multiplexer in a network node, along with the traffic and service models used in this research.

2.1 Network Model

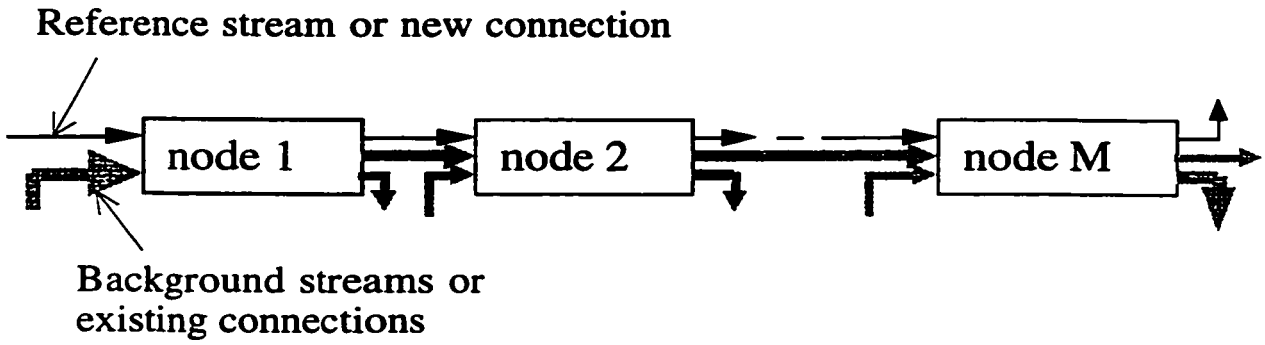


Fig. 2-1 The Network Model

The network environment used in our study is a tandem connection-oriented ATM network (Fig. 2-1), in which the RCS (Rate-Controlled Service) discipline is employed at every network node. The RCS discipline means that each traffic stream is reshaped at a traffic shaper / rate-controller before being delivered to the multiplexer at every node. As reviewed in section 1.1.8, this per-node traffic shaping discipline prevents the traffic flow from becoming more bursty as it

traverses multiple nodes. It also simplifies the task of analyzing the end-to-end performance that can be extended from the analysis for a single node.

The network nodes are assumed to be output queuing switches, and the switching delay is assumed to be negligible. The propagation delay is not considered in our research, since it does not affect the design of our traffic control schemes. We also assume a continuous flow model in our work, which means packetization time is not taken into account.

In our work on the end-to-end QoS provisioning, the traffic in the network of Fig. 2-1 is divided into the reference traffic and background traffic. There is one reference stream, and it traverses from the source node 1 to the destination node M. The other streams are called background streams. It is not necessary for a background stream to traverse from node 1 to node M, i.e., a background stream can leave the network after it departing from any node. The same comment applies to our work on admission control, except that we refer to them as the new connection (corresponding to the reference stream), and the existing connections (corresponding to the background streams) there.

The time in our network model is slotted. One time slot is the time needed to transmit one packet by an output link at its full capacity rate. In another word, the transmission capacity of the output link is one cell per time slot.

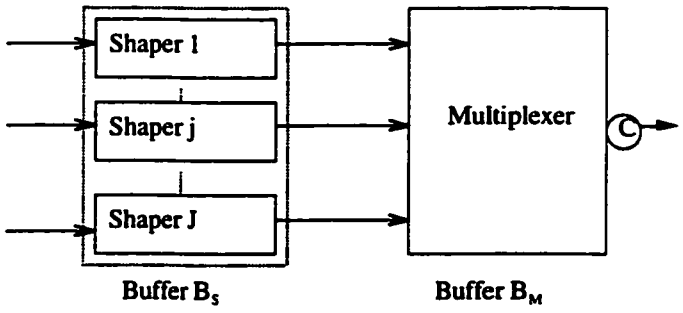


Fig. 2-2 Network Node Model

The network node model is illustrated using Fig. 2-2. For ease of exposition, we only draw one output port of the node in the figure. We can see that one multiplexer is available at every output link, and each incoming traffic stream is reshaped at the shaping buffer before entering the multiplexer. Thus, it can be seen that a network node applying the RCS discipline has two functions: rate-controlling and multiplexing. The rate-controlling function can be implemented by a multiple-leaky-bucket (MLB) traffic shaper.

2.2 Traffic Models

The following four traffic models are used in this thesis to characterize VBR traffic: the MLB traffic model, the DLB traffic model, the inter-arrival time process, and the background traffic model. Among them, the MLB / DLB traffic models and the inter-arrival time process are used to model an individual VBR traffic flow, while the background traffic model is used to model a bundle of VBR traffic streams.

2.2.1 The Multiple-Leaky-Bucket (MLB) Traffic Model

This multiple-leaky-bucket traffic model has been widely used to characterize the traffic in the broadband networks, since it can be easily enforced at the network entrance using a cascade of leaky buckets. When the *j*th traffic flow is characterized by the multiple-leaky-bucket regulated traffic model, it means that the number of arrivals at the network entrance from connection *j* during the time interval [s, s+t] can be upper-bounded by a multiple-leaky-bucket traffic constraint function $I_j^*(t)$, where *s* denotes an arbitrary point in the time evolution, and *t* represents the length of time interval. Essentially, $I_j^*(t)$ provides the upper bound of the number of arrivals from connection *j* during any time interval of length *t*.

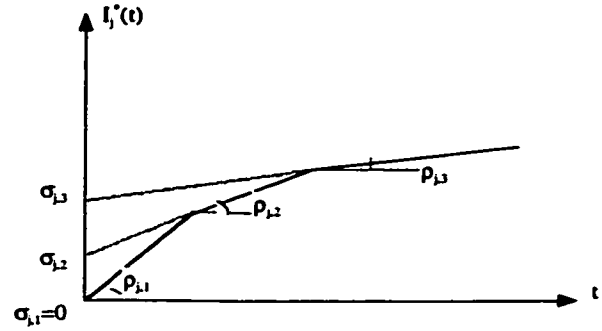


Fig. 2-3 An Illustration of $I_j^*(t)$

The multiple-leaky-bucket constrained function $I_j^*(t)$ is usually called the $(\bar{\sigma}, \bar{\rho})$ - constrained function, since it was shown in [Cruz91b] that $I_j^*(t) = \min_{1 \le n \le N_j}(\rho_{j,n}t + \sigma_{j,n})$, where $\bar{\sigma}_j = (\sigma_{j,1}, \ldots, \sigma_{j,n}, \ldots, \sigma_{j,N_j})$ is a vector denoting the y-intercepts of the line segments in $I_j^*(t)$, $\bar{\rho}_j = (\rho_{j,1}, \ldots, \rho_{j,n}, \ldots, \rho_{j,N_j})$ is a vector denoting the segment slopes of $I_j^*(t)$, and N_j is the number of segments. An example of $I_j^*(t)$ is illustrated in Fig. 2-3, in which $N_j = 3$. It is clear that the amount of arrivals during any time interval of length t is upper-bounded by $\min(\rho_{j,1}t, \rho_{j,2}t + \sigma_{j,2}, \rho_{j,3}t + \sigma_{j,3})$. Note that the X-axis value in Fig. 2-3 denotes the length of time interval instead of the time evolution.

The MLB traffic model will be used in chapter 3 to design an admission control algorithm. In order to simplify the computation of our admission control scheme, we use $(\bar{\rho}_j, \bar{\tau}_j)$ to describe $I_j^*(t)$, where $\bar{\tau}_j = (\tau_{j,1}, \ldots, \tau_{j,n}, \ldots, \tau_{j,N_j})$ is a vector denoting the X-axis values at the flex points of $I_j^*(t)$. The values of $\bar{\tau}_j$ can be derived from $(\bar{\sigma}_j, \bar{\rho}_j)$ - model as follows:

$\tau_{j,1} = \sigma_{j,1} / \rho_{j,1}$;

$\tau_{j,n} = (\sigma_{j,n} - \sigma_{j,n-1}) / (\rho_{j,n-1} - \rho_{j,n})$ for $n = 2, \ldots, N_j$.

The above traffic constraint function $I_j^*(t)$ can be a part of the traffic contract which is negotiated between the client and the network provider at the connection setup time. The MLB traffic model is easy to be enforced at the network entrance using a series of leaky buckets. However, the traffic pattern of connection *j* can be distorted when traversing even a single traffic multiplexer, which may lead to a more bursty traffic flow (e.g., a higher peak rate). Thus, the traffic model contract $I_j^*(t)$ may not be able to characterize the traffic flow *j* at the output port of any multiplexer or at the input port of any tandem or destination node. Therefore, in our analysis for a node inside the network, we use another MLB traffic constraint function $\tilde{I}_j^*(t) = \min_{1 \le n \le \tilde{N}_j} (\tilde{\rho}_{j,n} \cdot t + \tilde{\sigma}_{j,n})$ to characterize the incoming traffic flow *j*. $\tilde{I}_j^*(t)$ upper bounds the number of arrivals from connection *j* during the time interval [s, s+t], where $\tilde{\rho}_{j,n}$ denotes the slope of the *n*th line segment of $\tilde{I}_j^*$, $\tilde{\sigma}_{j,n}$ denotes the y-intercept of the *n*th segment of $\tilde{I}_j^*$, and $\tilde{N}_j$ is the number of line segments. Because of the concavity of $\tilde{I}_j^*$, we have $\tilde{\rho}_{j,n} > \tilde{\rho}_{j,n+1}$ and $\tilde{\sigma}_{j,n} < \tilde{\sigma}_{j,n+1}$, for $1 \le n \le \tilde{N}_j$. For ease of explanation, the incoming traffic flow *j* at a network node characterized by its MLB function $\tilde{I}_j^*(t)$ can be "imagined" to be output from a virtual MLB regulator with constraint function $\tilde{I}_j^*(t)$.

2.2.2 The Dual-Leaky-Bucket (DLB) Traffic Model

The dual leaky bucket (DLB) traffic model is a special case of the above MLB traffic model when the number of line segments in the MLB model is equal to two. The DLB traffic model has been adopted by the ATM Forum as the standard traffic model [ATMF96]. With the number of line segments in MLB traffic model equal to two, we let $\rho_{j,1} = P_j$, $\sigma_{j,1} = 0$, $\rho_{j,2} = \rho_j$, $\sigma_{j,1} = \sigma_j$. Then,

the DLB traffic model for connection j is denoted by (P_j, σ_j, ρ_j), where P_j bounds the peak rate, σ_j bounds the maximum burst size, and ρ_j bounds the sustainable cell rate. Then, the DLB traffic constraint function is $I_j^*(t) = \min(P_j t, \rho_j t + \sigma_j)$. We let $\tau_j = \sigma_j / (P_j - \rho_j)$, which is the maximum interval length during which the source can send cells in the peak-rate, and we refer to it as the "maximum peak-rate interval".

Similar to the MLB traffic model, the DLB traffic model is easy to be enforced at the network entrance. However, the traffic pattern of connection j can be distorted when traversing even a single traffic multiplexer. Thus, the DLB traffic constraint function $I_j^*(t)$ may not be able to characterize the traffic flow j at the output port of any multiplexer or at the input port of any tandem or destination node. In our analysis, we use the DLB traffic constraint function $\tilde{I}_j^*(t) = \min(\tilde{P}_j t, \tilde{\rho}_j t + \tilde{\sigma}_j)$ to characterize the incoming traffic flow j at a node inside the network, where $\tilde{P}_j$ upper bounds the peak cell rate of $\tilde{I}_j^*$, $\tilde{\rho}_j$ upper bounds its long-term-average-rate of $\tilde{I}_j^*$, and $\tilde{\sigma}_j$ upper bounds its burst size. For ease of explanation, the incoming traffic flow j at a network node characterized by its DLB parameters ($\tilde{P}_j$, $\tilde{\rho}_j$, $\tilde{\sigma}_j$) can be "imagined" to be output from a virtual DLB regulator with parameters ($\tilde{P}_j$, $\tilde{\rho}_j$, $\tilde{\sigma}_j$).

This worst-case traffic model is usually used to derive the deterministic performances of a system where lossless service is guaranteed. There is no statistical multiplexing gain in this case. On the other hand, when a small loss probability is allowed at the network node, the statistical multiplexing gain can be achieved by exploiting the assumed independent and bursty input traffic flows. The paper [MiMo95] provided an effective approach to derive the statistical QoS of the traffic flows from their deterministic DLB traffic model. It established that the estimation of loss probability given by Chernoff bound is maximized by a kind of extremal, on-

off, periodic processes with independent, uniformly distributed random phases. Based on their results, we assume in our analysis for the cell loss probability that the incoming traffic flows at a network node are extremal, on-off and periodic processes with independent and uniformly distributed random phases. Specifically, we assume that the token buffer of connection j's virtual regulator is full at the beginning of each session. The cells of connection *j* are sent at the peak-rate $\tilde{P}_j$ from its virtual regulator until the token buffer of the regulator becomes empty. The process will turn off as soon as the token buffer becomes empty, and it will not turn on until the token buffer becomes full. The incoming rate process of connection j, $\Omega_j(t)$, can be illustrated using Fig. 2-4, where $T_{on,j} = \tilde{\sigma}_j / (\tilde{P}_j - \tilde{\rho}_j)$, and $T_{off,j} = \tilde{\sigma}_j / \tilde{\rho}_j$. The duty cycle is $T_j = T_{on,j} + T_{off,j}$. The activity factor of the *j*th traffic flow is defined as $\omega_{\Omega_j} = T_{on,j} / T_j = \tilde{\rho}_j / \tilde{P}_j$.

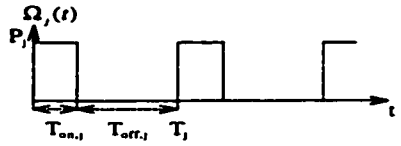


Fig. 2-4 Illustration of the Incoming Rate Process of Connection *j*

2.2.3 The Inter-arrival time process

Both MLB and DLB traffic models in the above two sub-sections are usually called the worst-case traffic models, because they characterize the upper-bound of the number of arrivals during certain time interval length. These two worst-case traffic models are used in our research on admission control and traffic shaping to derive the delay bound, the buffer occupancy bound, and the Chernoff estimation of loss probability. However when analyzing the probability distributions of the delay and delay jitter at a rate-controller or multiplexer, we need the inter-arrival time process, which is not characterized by the MLB / DLB traffic models.

In our analysis for the performance of a rate-controller, the inter-arrival time process is assumed to be generally and independently distributed with the probability distribution function $a(n)$ = Pr{inter-arrival time = n slots}. Let A_{min} (A_{max}) denote the minimum (maximum) possible value of the inter-arrival time. From the definition of time-slot in section 2.1, we know that $A_{min} \geq 1$. It is also assumed that $A_{max} < \infty$. Thus, we have $1 \leq A_{min} \leq n \leq A_{max} < \infty$. This traffic model was previously used in [LaSt97, OhMM91] to describe a single traffic flow in the ATM network.

Similarly, in our analysis for the performance of a FIFO multiplexer, the reference connection's inter-arrival time process is assumed to be generally and independently distributed with probability distribution function $\hat{a}(n)$= Pr{inter-arrival time = n slots}. The minimum (maximum) possible value of the inter-arrival time at a multiplexer is denoted by $\hat{A}_{\min}$ ($\hat{A}_{\max}$).

2.2.4 The background traffic model

A number of background traffic streams is introduced to interact with the reference stream at the multiplexer. There are N_b background streams input into a multiplexer, and they are assumed to be independent of each other. Each stream is modeled as a Bernoulli arrival process. That is, during one slot time, a cell arrives from a background stream with probability p, no cell arrives with probability $1 - p$. Thus, the background arrivals can be modeled as an i.i.d. batches, and the batch size Ψ_1 (number of arrivals in one slot) follows a binomial distribution $\psi_1(j) = Pr\{\Psi_1 = j\} = \binom{N_b}{j} \cdot p^j \cdot (1-p)^{N_b - j}$, for $0 \leq j \leq N_b$. Further, the batch size Ψ_n (the number of arrivals in n slots) follows a distribution $\psi_n(j) = Pr\{\Psi_n = j\} = \underbrace{\psi_1(j) \otimes \psi_1(j) \otimes \ldots \otimes \psi_1(j)}_{n}$, *i.e.*, $\psi_n(j)$ is n convolutions of $\psi_1(j)$. This type of background stream was first used in [OhMM91], and it was called the Markovian inter-arrival process due to the memory-less property of its inter-arrival

time. The reason why we choose this i. i. d. batch arrival model is because this model is able to characterize the burstiness of background traffic while allowing a simple analysis.

2.3 The MLB / DLB Traffic Shaper and its Operations

In our research, we shall use MLB for the shaper in Fig. 2-2. The MLB shaper is easy to implement by passing the traffic stream through a series of leaky buckets. It has been shown in [Cruz91b] that the number of departures from the MLB shaper of connection j is constrained by the shaper's envelope function $A_j(t) = \min_{1 \le n \le N_{Sj}} (\gamma_{j,n} \cdot t + \beta_{j,n})$, where $\gamma_{j,n}$ is the token rate of the nth leaky bucket, $\beta_{j,n}$ is the size of the token buffer in the nth leaky bucket, and N_{Sj} is the number of leaky buckets. The envelope function and the constraint function are used inter-changeably [Cruz91a, Chan94, GGPS96]. In this thesis, the traffic constraint function is used to describe the worst-case MLB / DLB traffic models, and we use the shaping envelope function to describe MLB / DLB shapers.

When the traffic flow j with the constraint function $\tilde{I}_j^*(t) = \min_{1 \le n \le \tilde{N}_j} (\tilde{\rho}_{j,n} \cdot t + \tilde{\sigma}_{j,n})$ is subject to shaping at the MLB shaper with envelope function $A_j(t) = \min_{1 \le n \le N_{Sj}} (\gamma_{j,n} \cdot t + \beta_{j,n})$, the following properties must hold: (1) Based on the property of traffic shaper, the peak shaping rate must be smaller than or equal to the peak rate of input traffic, i.e., $\gamma_{j,1} \le \tilde{\rho}_{j,1}$. (2) The system stability requires that the average shaping rate must be larger than or equal to the average input traffic rate, i.e., $\gamma_{j,N_{Sj}} \ge \tilde{\rho}_{j,\tilde{N}_j}$. (3) The upper-bound of connection j's shaping buffer occupancy $Q_{Sj} \le \tilde{I}_j^*(\tilde{\tau}_{j,\tilde{N}_j+1}) - \tilde{\rho}_{j,\tilde{N}_j} \cdot \tilde{\tau}_{j,\tilde{N}_j+1}$. (4) The upper-bound of the connection j's shaping delay $D_{Sj} \le \tilde{I}_j^*(\tilde{\tau}_{j,\tilde{N}_j+1}) / \tilde{\rho}_{j,\tilde{N}_j} - \tilde{\tau}_{j,\tilde{N}_j+1}$. Where, $\tilde{\tau}_{j,n} = (\tilde{\sigma}_{j,n} - \tilde{\sigma}_{j,n-1})/(\tilde{\rho}_{j,n-1} - \tilde{\rho}_{j,n})$ for $n = 1, \ldots, \tilde{N}_j$. When it

takes the equality in (3) and (4), we say that the input traffic is "fully shaped", and the envelope function of the shaper is $\tilde{\rho}_{j,N_j} t$. The above properties (3) and (4) can be easily obtained by applying the basic geometry to the graphs of both the traffic constraint function $\tilde{I}_j^*(t)$ and the shaping function $A_j(t)$.

The DLB traffic shaper is a special case of the MLB shaper when $N_{sj} = 2$. The DLB shaper of connection j has the envelope function $A_j(t) = \min(P_{sj} \cdot t, \gamma_j \cdot t + \beta_j)$, and it has three parameters: (1) P_{sj} bounds the peak rate of the shaped traffic. (2) β_j is the token buffer capacity which bounds the burst size of the shaped traffic. (3) γ_j is the long-term-average-rate. When the incoming traffic flow j with the traffic constraint function $\tilde{I}_j^*(t) = \min(\tilde{P}_j t, \tilde{\rho}_j t + \tilde{\sigma}_j)$ is subjected to shaping at the DLB shaper with envelope function $A_j(t) = \min(P_{sj} t, \gamma_j t + \beta_j)$, the system stability requires $\gamma_j \geq \tilde{\rho}_j$. We shall use $\gamma_j = \tilde{\rho}_j$ in this thesis.

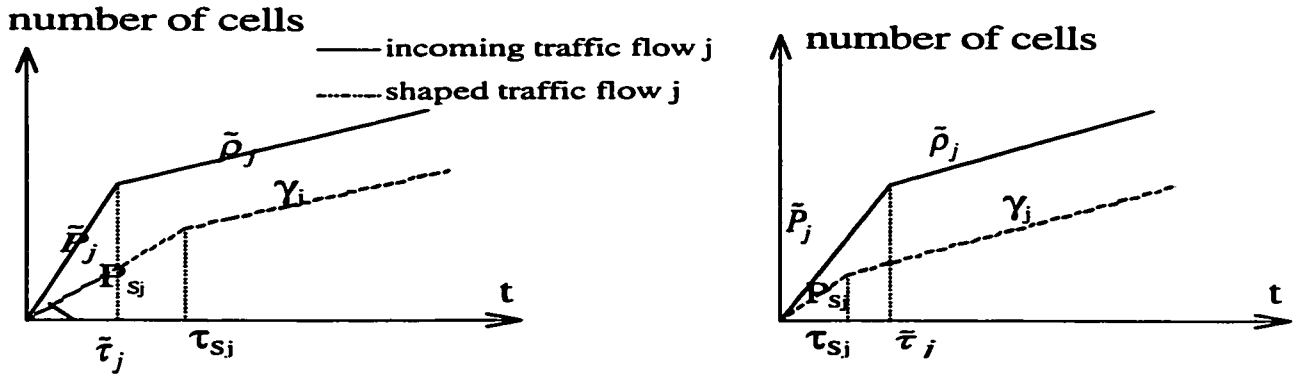
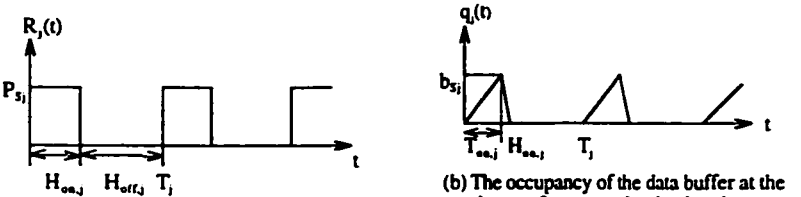


Fig. 2-5 Two Cases of DLB Traffic Shaper

Different from section 2.2.2 which is about the DLB traffic model of an incoming flow, this section discusses the traffic envelope functions of both the input and output traffic at a DLB shaper. When the traffic flow j with the traffic constraint function $\tilde{I}_j^*(t) = \min(\tilde{P}_j t, \tilde{\rho}_j t + \tilde{\sigma}_j)$ is

subjected to shaping at the DLB shaper with the envelope function $A_j(t) = \min(P_{sj}t, \gamma_j t + \beta_j)$, the traffic constraint functions of both the input and output flows at the shaper are illustrated in Fig. 2-5. In the figure, we let $\tilde{\tau}_j$ denote the maximum time interval length during which cells can be sent in the peak rate from the virtual regulator, then $\tilde{\tau}_j = \tilde{\sigma}_j/(\tilde{P}_j - \tilde{\rho}_j)$. Let τ_{sj} denote the maximum time interval length during which cells can be sent in the peak rate P_{sj} from the shaper, and we obtain $\tau_{sj} = \beta_j/(P_{sj}-\gamma_j)$.

Assume that the token buffer of the traffic shaper is full at the beginning of each session, then there are two different cases of the DLB shaper as shown in Fig. 2-5. In case (i), $\tilde{\tau}_j \leq \tau_{sj}$ which means that the shaper's token buffer cannot be emptied before the virtual regulator's token buffer. Let b_{sj} denote the maximum demand for the data buffer, then $b_{sj} = (\tilde{P}_j - P_{sj})\cdot\tilde{\tau}_j$. In case (ii), $\tilde{\tau}_j > \tau_{sj}$ which means that the shaper's token buffer has the potential to be emptied before the regulator's token buffer. In this case, $b_{sj} = (\tilde{P}_j - \gamma_j)\cdot\tilde{\tau}_j - (P_{sj} - \gamma_j)\tau_{sj}$. Obviously, the computation of b_{sj} is simpler in case (i) than in case (ii). We will use the shaper design of case (i) in Chapter 5. It is clear that all the results to be obtained in Chapter 5 can be extended easily to the shaper design of case (ii).



(a) The output rate process from the shaper of a connection in class j

(b) The occupancy of the data buffer at the shaper of a connection in class j

Fig. 2-6 Illustration of Processes $R_j(t)$ and $q_j(t)$ at the Shaper

Next, using the extremal, on-off, and periodic traffic model in section 2.2.2 for the incoming flows, we explain the shaper operation in our network node model of Fig. 2-2. We assume that J classes of traffic flows are input into the node, and that there are K_j connections in class j. All connections in class j are associated with the same set of DLB parameters $(\tilde{P}_j, \tilde{\sigma}_j, \tilde{\rho}_j)$, but they have different phases. That is, the incoming rate process of connection i in class j is $\Omega_{ji}(t) = \Omega_j(t + \theta_{ji})$, where $1 \le i \le K_j$, $1 \le j \le J$, and θ_{ji} is an independent random variable uniformly distributed in $[0, T_j]$. A little thought makes it clear that $R_{ji}(t)$, the output streams from the DLB shaper, are also extremal, on-off and periodic processes with random phases. Based on the assumption of independent input traffic flows, the processes $R_{ji}(t)$ of the same traffic class only differ in their phases, i.e., $R_{ji}(t) = R_j(t + \theta_{ji})$. $R_j(t)$ is shown in Fig. 2-6 (a). In the figure, $H_{on,j} = \beta_j / (P_{Sj}-\gamma_j)$ and $H_{off,j} = \beta_j / \gamma_j$. The activity factor of $R_j(t)$ is defined as $\omega_{R_j} = H_{on,j} / T_j = \gamma_j / P_{Sj}$, where $T_j = H_{on,j} + H_{off,j}$.

Let $q_{ji}(t)$ denote the instantaneous requirement of the shaping buffer from the ith connection of class j at time t. Since the traffic flows are mutually independent, we know that $q_{ji}(t) = q_j(t + \theta_{ji})$. The process $q_j(t)$ is illustrated in Fig. 2-6(b), where $b_{Sj} = (\tilde{P}_j - P_{Sj})T_{on,j}$.

2.4 The Operations of the Traffic Multiplexer

In our work on statistical traffic shaping, the multiplexer follows the resource allocation approach proposed in [ElMW95]. But unlike [ElMW95] in which traffic shaping is not considered, the input flows at our multiplexer are output from the traffic shapers.

At a traffic multiplexer with a buffer of capacity B_M and an output link of bandwidth C, the resources can be allocated in proportion to their respective capacities, i.e., $\frac{b_{Mj}}{B_M} = \frac{e_{0j}}{C}$, where

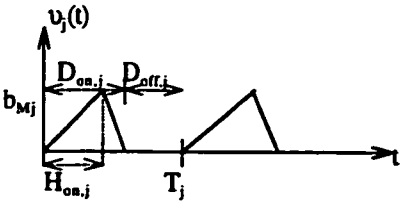
b_{Mj} denotes the multiplexing buffer allocation of connection j, and e_{0j} denotes the bandwidth allocation of connection j. Not only is this resource allocation approach simple in implementation, it was also established in [ElMW95] that both resources will always be exhausted at the same time if all connections use the same allocation rule. With this resource allocation approach, the traffic multiplexing function can be implemented by, for example, the ATDM (Adaptive Time Division Multiplexing) type of operation.

The following unique bandwidth allocation was obtained in [ElMW95] for connection j, which is characterized by a set of DLB parameters (P_{Sj}, β_j, γ_j).

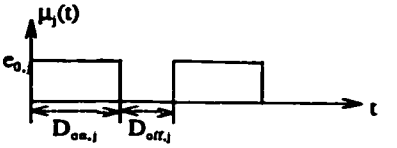
$$e_{0j} = \frac{P_{Sj}}{1 + \frac{B_M / C}{\beta_j}(P_{Sj} - \gamma_j)} \quad \text{if } \gamma_j \le \frac{\beta_j}{B_M / C}$$

$$= \gamma_j \quad \text{if } \frac{\beta_j}{B_M / C} \le \gamma_j \le P_{Sj} \tag{2-2}$$

Next, we look at the multiplexer operation, with the extremal, on-off, and periodic traffic processes $R_{ji}(t)$ input into the multiplexer,. At the multiplexer, the backlog of the ith connection in class j is served with bandwidth e_{0j} when it is not empty. Since the input traffic stream $R_{ji}(t)$ is an extremal, on-off and periodic process, the instantaneous buffer occupancy $\upsilon_{ji}(t)$ returns to zero in every duty cycle of length T_j. Based on the assumption of independent input traffic flows, we have $\upsilon_{ji}(t) = \upsilon_j(t + \theta_{ji})$. $\upsilon_j(t)$ is shown in Fig. 2-7 (a). In Fig. 2-7 (b), $\mu_j(t)$ denotes the utilized bandwidth at time t by a connection in class j. In [ElMW95], it was shown that $D_{on,j} = H_{on,j} + b_{Mj} / e_{0j}$, $D_{off,j} = H_{off,j} - b_{Mj} / e_{0j}$, and $D_{on,j}+D_{off,j}=T_j$. Finally, the fraction of time that this system is busy was obtained as $\chi_j = D_{on,j} / (D_{on,j} + D_{off,j}) = \gamma_j / e_{0j}$.



(a) The multiplexer buffer occupancy of a connection in class j



(b) The utilized bandwidth by a connection in class j

Fig. 2-7 Illustration of Processes $v_j(t)$ and $\mu_j(t)$ at the Multiplexer

2.5 Chapter Summary and Remarks

In this chapter, we have presented the network and traffic models. The network environment of our research is a tandem connection-oriented ATM network, in which the RCS discipline is employed at every node. Four traffic models are used in this thesis: the MLB traffic model, the DLB traffic model, the inter-arrival time process, and the background traffic model. Furthermore, we described the operations of both the traffic shaper and traffic multiplexer.

Chapter 3

An Admission Control Algorithm for Real-time VBR traffic

3.1 Introduction

We want to design an efficient bandwidth allocation scheme for the admission of real-time VBR service. The algorithm is designed for a connection-oriented ATM network (Fig. 2-1), in which the earliest deadline first (EDF) based rate-controlled service (RCS) discipline is employed at every network node. For ease of reference, we repeat the network model diagram in Fig. 3-1, and indicate in it the notations of both the delay bounds and the traffic constraint function.

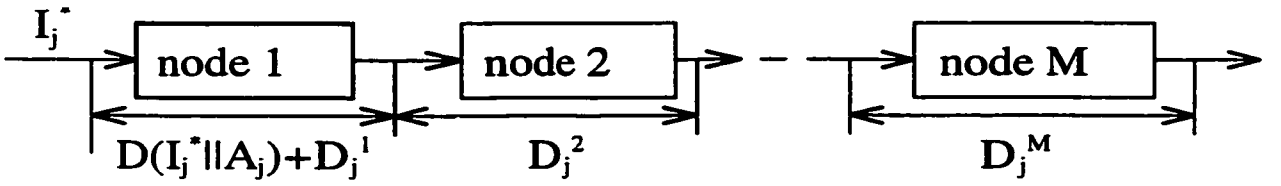


Fig. 3-1 Network Model

In the algorithm design, the *spare capacity curve* is introduced to characterize the available transmission capacity of the output link. The *service curve* used here is similar to the one in [Cruz95] where it is allocated to each connection. But unlike [Cruz95] which does not address the method of allocating the service curve, we present the approach to allocate the service curve and to update the spare capacity curve. Also, our algorithm does not need the traffic parameters of all the existing connections in every admission control in order to check the schedulability condition. By using the *spare capacity curve* to record the available bandwidth information, we only need the current spare capacity curve and the incoming connection's traffic parameters and QoS requirement in every admission control.

The multiple-leaky-bucket regulated traffic model is used in this chapter to characterize

the VBR traffic, which has been detailed in section 2.2.1. The computation of the delay bound in admission control can be simplified greatly by this assumption.

With reference to our network model in Fig. 3-1, we only consider in this chapter that identical shapers are used at each node on the route of a given connection. In this case, the end-to-end delay for connection j in our network model is bounded by

$$D_j = D(I_j^* \| A_j) + \sum_{m=1}^{M} D_j^m$$

where, D_j^m denotes the upper bound of the scheduling delay of connection j at node m, where the scheduling delay is defined as the duration between the time that the cell has completely arrived at the scheduler and the time that it is scheduled to be transmitted. I_j^* denotes the traffic constraint function of connection j at the network entrance. A_j stands for the envelope of the traffic shaper for connection j. $D(I_j^* \| A_j)$ represents the upper bound of the shaping delay, where the shaping delay is defined as the duration between the time that the cell has completely arrived at the shaper and the time that it is transmitted. We assume that the traffic is reshaped at every node to the same traffic constraint parameters as that at the network access point, i.e. $A_j = I_j^*$, thus, $D(I_j^* \| A_j)=0$. Therefore, the sufficient condition to admit a connection with end-to-end delay bound requirement of d_j is $\sum_{m=1}^{M} D_j^m \le d_j$.

3. 2 Definitions and Propositions

3.2.1 Definition of Service Curve

The service curve is allocated to each connection, and it is defined here as the upper bound of the service capacity that needs to be allocated to the new connection in order to guarantee the node delay bound.

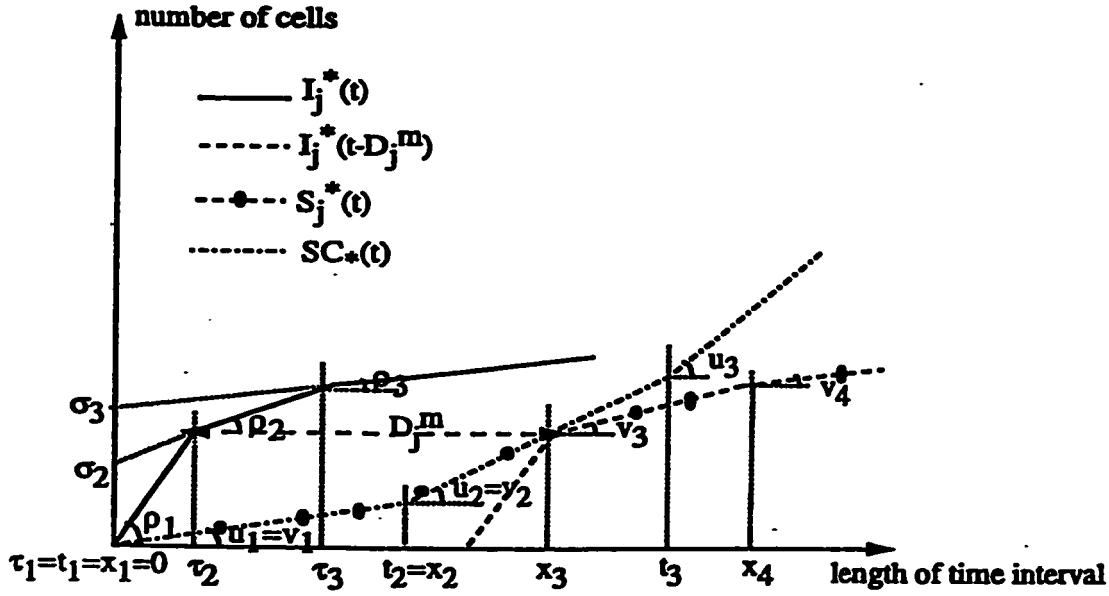


Fig. 3-2 An Illustration of $I_j^*(t)$, $S_j^*(t)$, and $SC_*(t)$

The definition of service curve can be interpreted using Fig. 3-2, in which D_j^m denotes the node delay bound of connection j at node m, and $S_j^*(t)$ represents the service curve allocated to connection j. The allocated $S_j^*(t)$ must be an upper-bound of the minimum number of cells from connection j that needs to be served during any time interval of length t in order to support D_j^m. Obviously, the minimum service needed is $I_j^*(t - D_j^m)$. Thus, $S_j^*(t)$ must be an upper bound of $I_j^*(t - D_j^m)$, i.e., $S_j^*(t) \geq I_j^*(t - D_j^m)$, for any $t \geq 0$. Since the upper bound of $I_j^*(t - D_j^m)$ is not unique, it is reasonable to assume that $S_j^*(t)$ is just a nondecreasing, nonnegative, and piecewise linear function. We shall use $\{(v_{j,l}, x_{j,l}) | l = 1, 2, ..., L_j\}$ to describe $S_j^*(t)$, where $x_{j,l}$ denotes the X-

axis value of the lth flex point in $S_j^*(t)$; $v_{j,l}$ is the slope of the lth segment; and L_j is the number of the line segments in $S_j^*(t)$. Thus, S_j^*(t) can be obtained as follows:

$$S_j^*(t) = v_{j,l}(t - x_{j,l}) + \sum_{k=1}^{l-1} v_{j,k}(x_{j,k+1} - x_{j,k}) \qquad \text{for } x_{j,l} \le t \le x_{j,l+1} \tag{3-1}$$

Fig. 3-2 shows an example of S_j^*(t) for $L_j = 4$.

As to be stated in the following Proposition 3.2, the computation of D_j^m can be simplified greatly by the assumption of convex spare capacity curve. Therefore, instead of allocating I_j^*(t - D_j^m) as the service curve of connection j, we allocate the service curve using the procedure to be presented in Proposition 3.3, which is designed to make the spare capacity curve convex.

3.2.2 Definition of Spare Capacity Curve

The spare capacity curve is used to characterize the available capacity of the output link. It is defined as the lower bound of the available service capacity at an output link. This definition can be explained using Fig. 3-2, in which $SC_*(t)$ represents the spare capacity curve at node m. That is, the number of cells that can be served by the available capacity of the output link during any time interval of length t is lower bounded by $SC_*(t)$.

Similar to the service curve, $SC_*(t)$ is assumed to be a nondecreasing, nonnegative, and piecewise linear function. In this chapter, we use $\{(u_h, t_h) | \text{ h=1, 2, ..., H}\}$ to describe $SC_*(t)$, where t_h denotes the X-axis value at the hth flex point in $SC_*(t)$, u_h is the slope of the hth segment, and H is the number of the line segments. In Fig. 3-2, H = 3, and $SC_*(t)$ can be obtained as

$$SC_*(t) = u_h(t - t_h) + \sum_{k=1}^{h-1} u_k(t_{k+1} - t_k) \quad \text{for } t_h \le t \le t_{h+1} \tag{3-2}$$

As a trivial case, in our discrete-time network model, $SC_{\bullet}(t)$ = t when there is no connection traversing the output link. This means that one slot is normalized to the time needed to transmit one cell at its full capacity rate, as defined in section 2.1.

In our algorithm, D_j^m the minimum upper bound of the node delay is computed as the maximum horizontal distance between $I_j^{\bullet}$(t) and $SC_{\bullet}(t)$. Thus, the admission control based on this delay bound may reject some connections which could be admitted if the delay bound is computed using actual service capacity. However, the delay bound computation can be rather complicated by using the actual service capacity. As to be detailed in the following propositions 3.2 and 3.4, the computation of delay bound is very efficient by using the convex lower bound of the spare capacity curve.

3.2.3 Propositions on Bandwidth Allocation

This section presents a few propositions on bandwidth allocation that make use of the service curve and the spare capacity curve definitions introduced earlier. These propositions provide the analytical foundation of our admission control algorithm in section 3.3.

The following assumptions will be used throughout the rest of this chapter.

(1) The input stream *j* is assumed to be conforming to a piecewise linear, non-decreasing, and concave traffic constraint function $I_j^{\bullet}$ described by $\{(\rho_{j,n}, \tau_{j,n}) | n = 1, ..., N_j\}$ [WKZL96].

(2) The Request message of connection *j* arriving at node *m* includes the information $I_j^{\bullet}$, d_j, and $\bar{D}_j^m$, where, d_j is the end-to-end delay bound requirement of connection *j*, and $\bar{D}_j^m$ is a vector recording the minimum scheduling delay bounds incurred at those upstream nodes of node *m*.

(3) The Accept message of connection *j* consists of the vector $\bar{\hat{D}}_j^m$ which records the reassigned node scheduling delay bounds on the route of connection *j*.

(4) The connection Requests are processed one by one at a network node. That is, the node will not process the next connection Request until the presently processed one has been either accepted or rejected. This implies that the multi-threaded software technique may not be used to implement the proposed admission control algorithm.

(5) The spare capacity curve will not be updated and the bandwidth allocation will not be reserved until the Accept message is received.

Proposition 3.2 provides the computation of the minimum upper bound of the scheduling delay under the assumption of the convex spare capacity curve. The assumption of the convex spare capacity curve is made on the basis of the concavity of the traffic constraint function. It is established in Proposition 3.2 that this assumption can greatly simplify the computation. Proposition 3.3 gives the procedure to compute the service curve in order to guarantee the delay bound. In the design of Proposition 3.3, we try to keep the spare capacity curve piecewise linear and convex, so that the computation of the scheduling delay bound can be simplified. However, the updated spare capacity curve can not always remain convex when the bandwidth is allocated using the reassigned value of the node delay bound, as to be detailed before Proposition 3.4. Therefore, we suggest to use the convex lower bound of the spare capacity curve when computing the node delay bound, and the reason behind this suggestion is given in Proposition 3.4. The procedure to update the spare capacity curve is put in Proposition 3.1, since it will be used in the proof of Proposition 3.3.

Proposition 3.1: Procedure to update spare capacity curve

Let $SC_{\bullet}(t)$ be the spare capacity curve at certain output link right before the connection j is admitted, and let $S_j^{\bullet}(t)$ denote the service curve allocated to the connection j. Then, right after the connection j is admitted, the updated spare capacity curve $SC'_{\bullet}(t)$ can be obtained as $SC'_{\bullet}(t) = SC_{\bullet}(t) - S_j^{\bullet}(t)$.

Proof: see Appendix A.1.

Proposition 3.2: Procedure to compute the minimum upper bound of the scheduling delay

Assume that the current spare capacity curve denoted by $\{(u_h, t_h) | h = 1, ..., H\}$, is a piecewise linear function, convex during $[0, T_B]$, where T_B is the maximum length of the busy period of the output link. Let d_j^m denote the scheduling delay to be incurred on the incoming connection j at node m. We have

(1) If $u_H < \rho_{j,N_j}$, d_j^m can not be upper bounded by any constant $D_j^m < \infty$.

(2) If $u_H \geq \rho_{j,N_j}$, the minimum upper bound of d_j^m is given by

$$D_j^m = \max_{t:t\geq 0} \min\{\delta t: \delta t \geq 0 \text{ and } I_j^*(t) \leq SC_*(t+\delta t)\}. \quad (3\text{-}3)$$

Proof: see Appendix A.2.

Example of Proposition 3.2:

When $N_j = 2$, there are four cases in computing $D_j^m < \infty$. Fig. 3-3 gives the geometrical interpretation of these four cases in computing $D_j^m < \infty$ and of the case when $D_j^m = \infty$. Let u_f denote the slope of $SC_*(t)$ at the point where $SC_*(t) = \rho_{j,1} \cdot \tau_{j,2}$.

case (i) If $\rho_{j,1} \leq u_f$, $D_j^m = d_j^m = 0$.

case (ii) If the peak rate of the new connection $\rho_{j,1} < u_f$, d_j^m is upper bounded by $D_j^m = t_h - SC_*(t_h)/\rho_{j,1}$, where t_h is a beginning point of the hth segment in $SC_*(t)$, $u_h < \rho_{j,1}$, and $u_{h+1} \geq \rho_{j,1}$.

case (iii) If $\rho_{j,2} \leq u_f \leq \rho_{j,1}$, d_j^m is upper bounded by $D_j^m = SC_*^{-1}(\rho_{j,1} \cdot \tau_{j,2}) - \tau_{j,2}$. Where, SC_*^{-1} is the inverse function of $SC_*(t)$ in equation (3-2).

case (iv) If $u_f < \rho_{j,2}$, d_j^m is upper bounded by $D_j^m = t_h - \tau_{j,2} - (SC_*(t_h) - \rho_{j,1} \cdot \tau_{j,2}) / \rho_{j,2}$, where t_h is the beginning point of hth segment in $SC_*(t)$, $u_h < \rho_{j,2}$, and $u_{h+1} \geq \rho_{j,2}$.

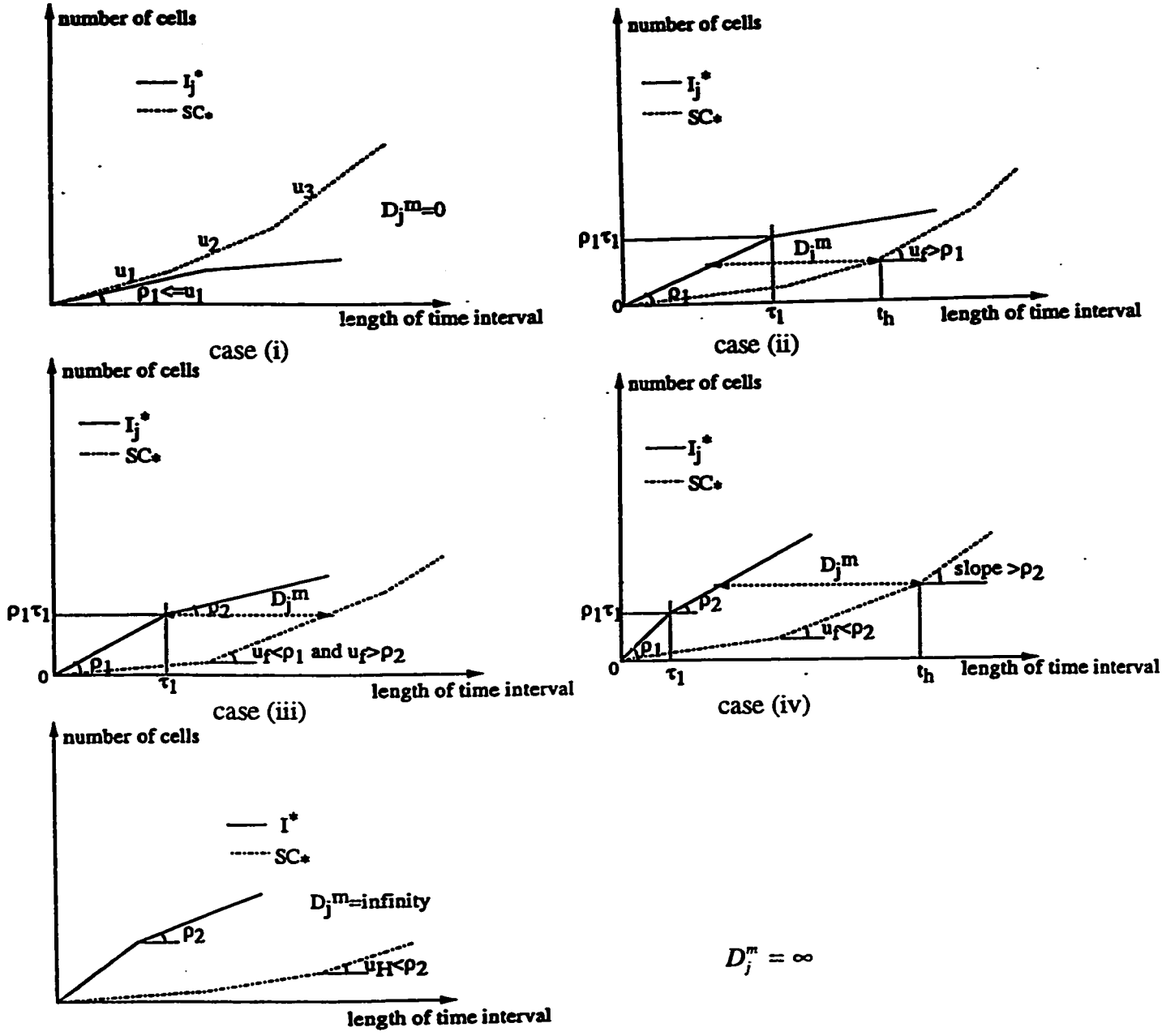


Fig. 3-3 Various Cases of $D_j^m < \infty$ and $D_j^m = \infty$ (when $N_j = 2$)

From Proposition 3.2, we can see that the computation of D_j^m is really simple under the assumption of convex spare capacity curve. It is clear that D_j^m is the upper bound of the scheduling delay, since $SC_\bullet(t)$ is the lower bound of the available link capacity and $I_j^\bullet(t)$ is the traffic constraint function. Thus, the admission control based on D_j^m may reject some connection which could be admitted based on the actual cell delay. However, the computation of actual cell delay can be very complicated, so the delay bound is often used in the admission control.

After D_j^m is computed, the admission control will allocate the service curve $S_j^\bullet(t)$ to connection j using the following proposition.

Proposition 3.3: Procedure to allocate the service curve

Obtained by using Proposition 3.2, the minimum upper bound on the node scheduling delay D_j^m can be guaranteed by using the following two rules to allocate the service curve:

Rule (1): If $D_j^m = 0$, the service curve allocated to connection j will be the same as its traffic constraint function, i.e. $S_j^\bullet(t) = I_j^\bullet(t)$.

Rule (2): If $0 < D_j^m < \infty$, the service curve will be allocated as: $S_j^\bullet(t) = SC_\bullet(t)$ for $0 < t < t_d$, and $S_j^\bullet(t) = I_j^\bullet(t - D_j^m)$ for $t \geq t_d$,

where, t_d is the minimum X-axis value of the point in $SC_\bullet(t)$ from which the horizontal distance to $I_j^\bullet(t)$ is equal to D_j^m, i.e. $t_d \overset{def}{=} \min_{t \geq D_j^m} \{t : SC_\bullet(t) = I_j^\bullet(t - D_j^m)\}$.

Furthermore, the updated spare capacity curve after connection j is admitted can remain as a piecewise linear function, convex during $[0, T_B]$.

Proof: see Appendix A.3.

Remarks:

(i) When allocating the service curve or updating the spare capacity curve during admission control, we do not need to obtain the function $S_j^*(t)$ or $SC_*'(t)$, we only need to compute the segment slopes using the simple subtraction operation.

(ii) t_d can be obtained at the same time when D_j^m is computed using Proposition 3.2.

It is easy to understand the purpose of Rule (1), but one may question Rule (2). There are two concerns when designing Rule (2). The first is that we try to keep the updated spare capacity curve piecewise linear and convex when allocating $S_j^*(t)$, since Proposition 3.2 shows that the assumption of the convex spare capacity curve makes it very efficient to compute D_j^m. The second is that the computation complexity increases with the increasing number of flex points in $SC_*(t)$, so Rule (2) is designed to remove those flex points that do not help to admit more connections. According to Rule (2), $SC_*'(t_d) = 0$ after the new connection is admitted, so it does not help to keep such flex points as (t_e, y_e) in $SC_*(t)$ where $t_e < t_d$ and $y_e > 0$, since D_j^m is computed by exploiting the convexity of $SC_*(t)$.

At the destination, the cumulative scheduling delay bound $accud_j = \sum_{m=1}^{M} D_j^m$ is calculated. If it is less than d_j, the node delay bounds can be less tight than the values computed using Proposition 3.2, and less service capacity is needed to support d_j. Let $\hat{D}_j^m$ denote the reassigned value of the D_j^m. Since the method to reassign $\hat{D}_j^m$ is not the concern of this work, we simply choose the node with the largest D_j^m and reassign $\hat{D}_j^m = D_j^m + (d_j - accud_j)$ for that node, and keep $\hat{D}_j^m = D_j^m$ for the other nodes. The vector $\hat{D}_j^m$ is sent with the Accept message to the source.

The service curve is reallocated as $S_j^*(t-\delta D_j^m)$ where $\delta D_j^m = \hat{D}_j^m - D_j^m$. It is easy to see that the updated spare capacity curve can not always remain convex when the service curve is reallocated as $S_j^*(t-\delta D_j^m)$. One may think that tests are needed before reassigning $\hat{D}_j^m$ in order to keep the spare capacity curve convex, which will increase the computation complexity of our algorithm. However, the major benefit of having a convex spare capacity curve is to make the admission control more efficient. Thus, instead of keeping the spare capacity curve strictly convex, we suggest to construct a convex lower bound of the spare capacity curve before computing D_j^m, as illustrated in Fig. 3-4. The reason behind this suggestion is given in Proposition 3.4.

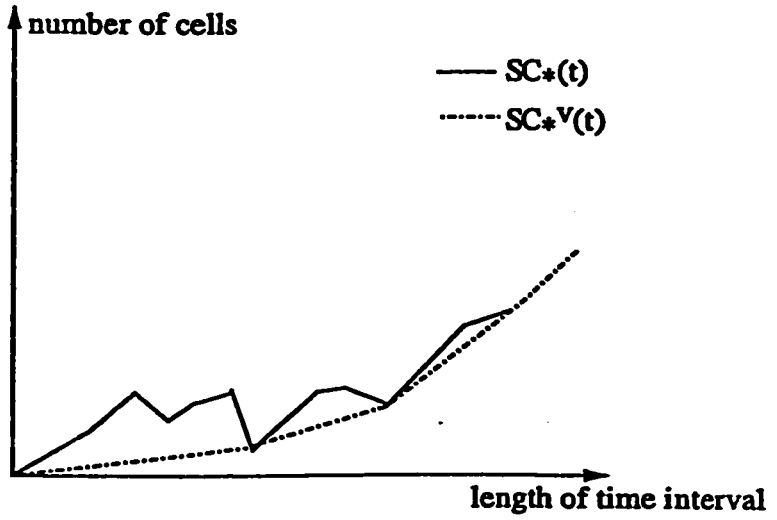


Fig. 3-4 The Spare Capacity Curve and its Convex Lower Bound

Proposition 3.4: Using the convex lower bound of the spare capacity curve to compute the node delay bound

Let $SC_*^v(t)$ denote a convex lower bound curve of $SC_*(t)$, and it is described by $\{(u_h^v, t_h^v) \mid h = 1,\ldots,H^v\}$. If d_j^m can be upper-bounded by any constant $D_j^m < \infty$, then it is upper-bounded by the minimum scheduling delay bound computed using $SC_*^v(t)$.

Proof: see Appendix A.4.

From the above proposition, we see that it is reliable to do the admission based on the convex lower bound of the spare capacity curve. The admission control based on $SC_*^v(t)$ may reject some connection request which can be admitted if the control is based on $SC_*(t)$. However, constructing the convex lower bound curve of the spare capacity is only needed when computing D_j^m, and the spare capacity information is not changed.

Thus far, we can see that the use of the service curve and the spare capacity curve in our bandwidth allocation approach simplifies the node delay bound computation and thereby reduces the computational time needed for the admission control. First, the spare capacity curve is used to record the available bandwidth, therefore only the new connection's traffic parameters are needed in the admission control. In those previous schemes, all the existing connections' traffic parameters are needed in one admission control to check the schedulability condition. Second, the service curve allocation procedure presented in Proposition 3.3 is designed to make the spare capacity curve convex. As shown in Propositions 3.2 and 3.4, the computation of the node delay bound can be simplified greatly by using the convex lower bound of the spare capacity curve.

3.3 Admission Control Algorithm

Based on the analysis in section 3.2, we present the pseudocodes of our admission control algorithm in the following Fig. 3-5. The algorithm consists of two major procedures: *fwdcontrl* and *bkwdcontrl* . Procedure *fwdcontrl* is designed based on Propositions 3.2, 3.3, and 3.4, and it is invoked by the Request message from the source node. Procedure *bkwdcontrl* is based on Propositions 3.1 and 3.3, and it is invoked by the Accept message from the destination.

Procedure *fwdcontrl*$(I_j^*, d_j, \bar{D}_j^m, \hat{D}_j^m, SC_*, S_j^*)$
  if $I_j^*(t) \le SC_*(t)$ for all $t = t_h$ **then**
    $D_j^m = 0$;

```
    alloc_serv(I_j^*, SC_*, S_j^*, D_j^m);
  else
    convexsc(SC_*, SC_*^v);
    nd_delay(I_j^*, SC_*^v, D_j^m);
    alloc_serv(I_j^*, SC_*, S_j^*, D_j^m);
  end if
  if (Σ_{k=1}^{m} D_j^k > d_j) then
    send Reject message to the source;
  else
    if (this is the destination node) then
        d_extra = d_j − Σ_{m=1}^{M} D_j^m;
        if (d_extra >0) then
            maxi=max(D̄_j^m);
            for m=1 to M do
               if m=maxi then D̂_j^m = D_j^m + d_extra;
                         else D̂_j^m = D_j^m;
               end if
            end for
        else
            for m=1 to M do
               D̂_j^m = D_j^m;
            end for
        end if
        send Accept message to the source;
    else
        send Request message to the next node;
    end if
  end if

Procedure bkwdcontrl(I_j^*, D̄_j^m, D_j^m, SC_*, Ŝ_j^*, S_j^*)
  if (D̂_j^m > D_j^m) then
    realloc_serv(I_j^*, D_j^m, D̂_j^m, S_j^*, Ŝ_j^*);
    updatesc(Ŝ_j^*, SC_*);
  else
    updatesc(S_j^*, SC_*);
  end if
```

Fig. 3-5 Pseudocodes for the CAC Algorithm

When invoked by a Request message of certain new connection, procedure *fwdcontrl* will find out whether the performance requirement of the new connection can be guaranteed without affecting the QoS of the existing connections. As shown in Fig. 3-5, function *convexsc* is used to construct $SC_\bullet^v$ the convex lower bound of the current spare capacity curve, and then function *nd_delay* is invoked to compute D_j^m based on $SC_\bullet^v$. Function *alloc_serv* computes the service curve $S_j^\bullet$ using Proposition 3.3. If the cumulative delay up to this node exceeds end-to-end delay bound, a Reject message will be sent to the source; otherwise, the Request message will continuously be sent to the destination. At the destination node, if the cumulative scheduling delay is less than the end-to-end delay bound requirement, the function *max* is used to obtain the node at which D_j^m is the largest on the route, and it randomly chooses one if more than one node has the same largest value of D_j^m. Then, the scheduling delay bound at the node with the largest D_j^m is reassigned as $\hat{D}_j^m = D_j^m + (d_j - accud_j)$, and the delay bound at other nodes are not changed.

At the beginning of *fwdcontrl*, we check whether the envelope $I_j^\bullet$ indicated in the Request message can be upper bounded by $SC_\bullet$ before computing the convex lower bound of $SC_\bullet$, which helps to reduce the number of connections rejected unnecessarily.

When invoked by an Accept message of the new connection, procedure *bkwdcontrl* is responsible for reserving the bandwidth for the new connection. If the scheduling delay bound D_j^m is not reassigned, function *updatesc* is called to update the spare capacity curve $SC_\bullet$ using Proposition 3.1; otherwise, function *realloc_serv* needs to be used to reallocate the service curve $\hat{S}_j^\bullet$ according to the reassigned $\hat{D}_j^m$, and function *updatesc* is then used to update the spare capacity curve based on $\hat{S}_j^\bullet$.

In reallocating the service curve, as shown in Fig. 3-6, we try to avoid adding many flex points in the updated SC_* which may make the admission control less efficient.

Procedure ***realloc_serv***$(\bar{D}_j^m, D_j^m, \hat{S}_j^*, S_j^*)$

$\delta d_j^m = \hat{D}_j^m - D_j^m$;

$S'(t) = S^*(t - \delta d_j^m)$; /* shift $S^*(t)$ to the right for a distance of δd_j^m */

Find points $(x_{j,l}, S'(x_{j,l}))$ on the $S'(t)$ curve;

Connecting the points found in the order of the time values, obtain $\hat{S}_j^*(t)$;

Fig. 3-6 Pseudocodes for the *realloc_serv* procedure

When a connection is disconnected, its reserved service capacity must be returned, and the spare capacity information always needs to be updated. Assume that its allocated S_j^* is described by $\{(v_{j,l}, x_{j,l}) | \ l = 1, ..., L_j\}$, and SC_* is denoted by $\{(u_h, t_h) | \ h = 1, ..., H\}$ before the disconnection, then the updated spare capacity curve SC_*' is given by $\{(u_h', t_h) | \ h = 1, ..., H\}$, where $u_h' = u_h + v_{j,l}$ for $x_{j,l-1} < t_h \le x_{j,l}$.

3.4 Performance Evaluation

For an approach to be used as an admission control for the delay-sensitive VBR traffic, it must be fast enough to fit the real-time requirement and able to make efficient use of the network resource. Therefore, we carried out some simulations to evaluate our admission control algorithm in terms of the computation time needed to admit one connection and the maximum number of admissible connections. The simulation is event-driven. Every time a new connection needs to be set up, the new connection's traffic parameters and its end-to-end delay bound requirement are input into the simulation program. The simulation is implemented in C programming language, and it has about 450 lines of code. It computes the delay bound based on the input information of

the new connection and the spare capacity curve, and it makes the decision on whether to admit or reject the new connection.

In the experiments below, we consider an ATM network with 5 nodes in tandem, in which the output link capacity is 45 Mb/s at every node. Two real video traces are used in our experiments, which are provided by O. Rose [Rose95]. We assume that each frame is packetized into 48-byte ATM cells. The first trace is called *terminator*, which has the peak rate of 1.6 Mb/sec, the burst size of 2929 cells, and the long-term-average-rate of 320 Kbits/sec. The parameters for the second trace are derived from the video sequence called *dino* taken from the movie Jurassic Park; the peak rate is 2.68 Mb/s, the burst size is 17203 cells, and the long-term-average-rate is 370 Kb/sec. As assumed in section 3.3, these parameters will be included in the Request message.

Fig. 3-7 shows the characteristics of these two traces by using the curves of the worst-case average rate versus the time interval length. The worst-case average rate is computed as the (maximum number of bits generated during an arbitrary time interval)/(time interval length). Thus, Fig. 3-7 depicts essentially the upper bound of the average rate over various arbitrary time intervals of different length. This figure also reflects how quickly the bounding average rate converges towards the long-term-average-rate. We can see that the bounding average rate of trace *term* converges more quickly than that of trace *dino*, which indicates that the trace *dino* is more bursty than the trace *term*.

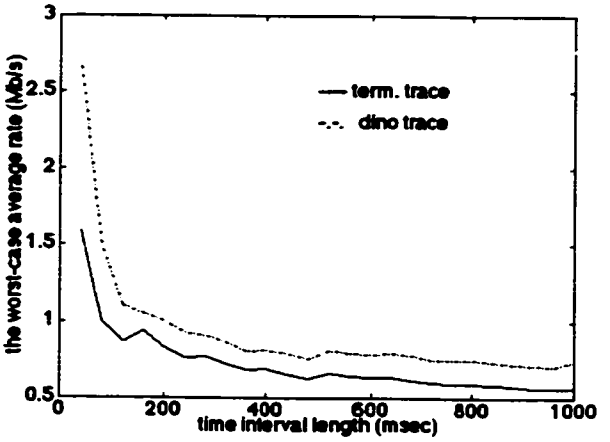
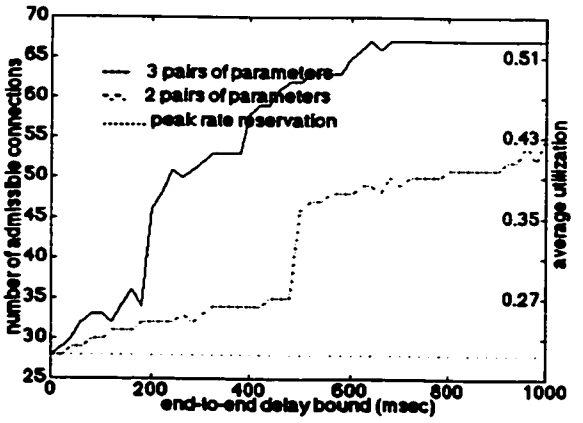
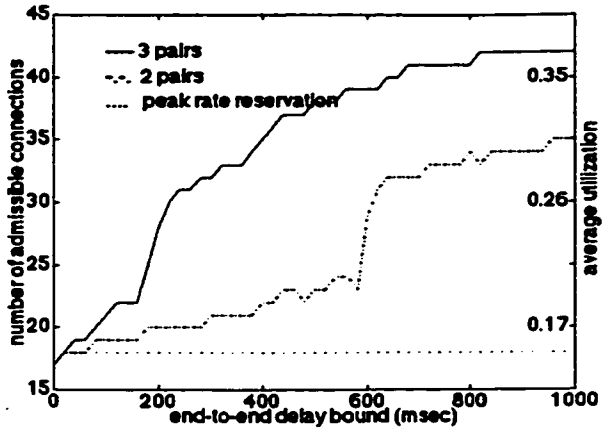


Fig. 3-7 Traffic Characteristics of Two Video Traces

In the first experiment, we obtain the maximum number of homogeneous connections that can be multiplexed on an end-to-end path so that all connections receive a deterministic guarantee on end-to-end delay. The average utilization is calculated as $N_d \cdot avg$ /(link capacity), where N_d is the number of connections admitted when the end-to-end delay bound is d, and avg is the long-term-average-rate.



(a) trace *terminator*



(b) trace *dino*

Fig. 3-8 Network Utilization vs. the End-to-End Delay Bound

There are several noteworthy points about Fig. 3-8:

(1) our algorithm is significantly better than the peak-rate reservation algorithm, and can achieve very reasonable utilization when the deterministic QoS is provided. For example, when three $(\bar{\rho}, \bar{\tau})$ pairs are used and the end-to-end delay bound of 300 msec must be guaranteed (i.e., the average node delay bound is 60 msec), we can achieve the average utilization of 0.42 for the *terminator* trace and the utilization of 0.30 for the *dino* trace. Comparing with the results in [WKZL96] where the EDF scheduler and the trace *dino* were also used, we can see the average utilization achieved here is higher than that achieved through checking the schedulability condition.

(2) The utilization is not strictly increasing with the end-to-end delay bound. However, the predominant trend is that the utilization increases with the increasing end-to-end delay bound, and the oscillation is mainly due to two functions *convexsc* and *realloc_serv* in our algorithm.

(3) Just as expected, larger utilization can be achieved when the arrival traffic is described by three $(\bar{\rho}, \bar{\tau})$ pairs than that when traffic is described by two $(\bar{\rho}, \bar{\tau})$ pairs, which means that the

maximum achievable utilization is affected by both the traffic model and the admission control algorithm.

(4) The achievable utilization for a given delay bound is lower for the trace *dino* than for the trace *terminator*. This is because the rate of the trace *dino* converges slower to the long-term-average-rate than that of the trace *terminator*.

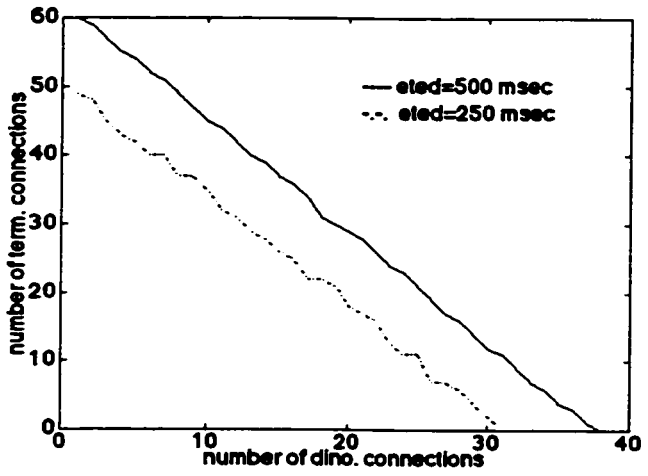


Fig. 3-9 Maximum Number of Admissible Connections

In the second experiment, we show that our algorithm is able to accommodate heterogeneous connections and QoS requirements. Fig. 3-9 depicts the admissible region of multiplexing combinations of the above two types of traces. The lower curve shows the maximum number of the *dino* and *terminator* connections can be multiplexed so that the packets from all connections can be guaranteed an end-to-end delay bound of 250 msec. The upper curve represents the maximum admissible region for a 500 msec end-to-end delay bound.

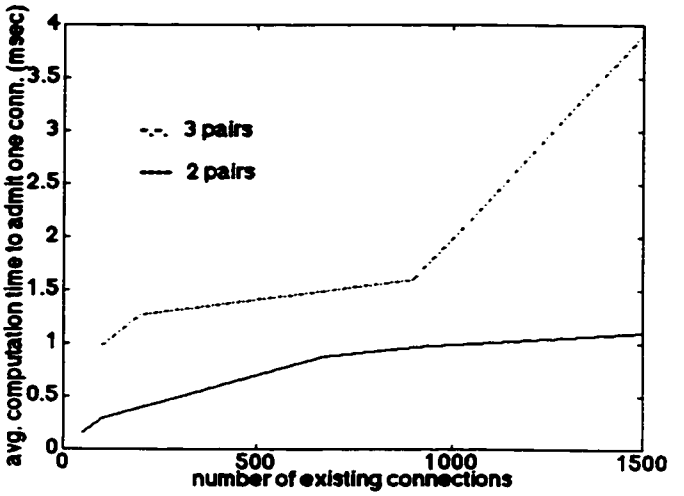


Fig. 3-10 Computation Time vs. Number of Existing Connections

The third experiment evaluates the computation time needed to admit one connection. The admission control program was run on a SUN Sparc-5 workstation with a 110 MHz micro SPARC-II processor and 64MB RAM in it. Fig. 3-10 shows the average CPU time needed to admit one connection versus the number of existing connections for the two cases when the incoming traffic constraint function is described by two and three (ρ, τ) pairs respectively. Comparing with the results in [FiKT97] where the traffic is described by two (ρ, τ) pairs, the computation time needed using our algorithm is shorter than that is needed when admission control is done through checking the exact schedulability condition. In our algorithm, procedure *nd_delay* calculates the minimum upper bound of the scheduling delay by exploiting the convex lower bound of the spare capacity curve, and the procedure *alloc_serv* removes the flex points in the spare capacity curve that do not help to admit more traffic. These two designs help to reduce the computation time.

Fig. 3-8 showed that higher utilization can be achieved by using three (ρ, τ) pairs than that by using two (ρ, τ) pairs. However, it can be seen from Fig. 3-10 that the computation time needed when the traffic is described by three (ρ, τ) pairs is much longer than that when the traffic is described by two (ρ, τ) pairs, so there is a trade-off between the computation time and the network utilization. The computation time plotted in Fig. 3-10 can be further reduced by

using the classification method in [FiKT97], where the delay requirements are classified in L_C classes and the computational complexity is reduced to $O(L_C)$.

Next, we are going to provide two tables to compare the performance of our algorithm with that of the previous works. As mentioned in section 1.1.2, three major kinds of real-time VBR admission control algorithms have been discussed in the past works. The approach of checking schedulability condition [FeVe90, ZhSh94, WKZL96, LiWF96, KnZh97], the approach of classifying the delay bound requirements [FiKT97], and the peak rate reservation approach [WKZL96, KnZh97].

Algorithms	Bandwidth Utilization		
	end-to-end delay bound = 250 msec	end-to-end delay bound = 500 msec	end-to-end delay bound = 1000 msec
our algorithm	0.27	0.32	0.36
checking schedulability	0.22	0.28	0.3
classifying delay bounds	< 0.22	< 0.28	< 0.3
peak rate reservation	0.1	0.1	0.1

Table 3-1 Comparison Results of Bandwidth Utilization

The comparison of bandwidth utilization performance is provided in Table 3-1. Since it was shown in both [WKZL96] and our first experiment that more than two (σ, ρ) pairs must be used to characterize traffic flows in order to achieve reasonable bandwidth utilization, we use three (σ, ρ) pairs for our comparison. In the table, the values of bandwidth utilization for the approach of checking schedulability condition is obtained from [WKZL96], which used the same EDF scheduler and the same movie trace *dino* as in this thesis. Note that the delay bound values used in [WKZL96] are node delay bounds, and that the delay bound values used in our Fig. 3-8 are end-to-end delay bounds. Therefore, in our comparison, their end-to-end delay bound values are estimated by multiplying their node delay bounds with the number of nodes. The bandwidth utilization performance of the classifying delay bounds approach was not analyzed in [FiKT97], but it was claimed in [FiKT97] that their algorithm has the drawback of admitting fewer

connections than the approach of checking the schedulability condition. It can be seen from Table 3-1 that, for various values of end-to-end delay bound, our algorithm can achieve higher bandwidth utilization than the previous algorithms.

Algorithms	Average computation time to admit one connection		Complexity	Operation
	number of existing connections = 1000	number of existing connections = 1500		
our algorithm	0.95 msec	1.1 msec	$O(N_f)$	subtraction
checking schedulability	1.3 msec	2.0 msec	$O(N)$	addition, subtraction, multiplication
classifying delay bounds	0.01 msec	0.01 msec	$O(L_c)$	addition, subtraction, multiplication.
peak rate reservation	CPU time to conduct one division		$O(1)$	one division

Table 3-2 Comparison Results of Computation Time, Complexity, and Operation

Table 3-2 compares the performances of the average computation time to admit one connection, the computational complexity, and the operation type. The performance of the average computation time was provided in [FiKT97] for both the method of checking the exact schedulability condition and the method of classifying the delay bounds. They used two (σ, ρ) pairs in [FiKT97]. It can be seen from Table 3-2 that computation time of our algorithm is much shorter than that of the checking schedulability approach, but it is longer than the method of classifying delay bounds. However, the method of classifying delay bounds achieves lower bandwidth utilization than ours. We know from Proposition 3.3 that the computational complexity of our algorithm is $O(N_f)$, where N_f is the number of flex points on the spare capacity curve. We let N denote the number of existing connections. It is easy to know from our Proposition 3.3 that $N_f < N$. Thus, our algorithm has lower complexity than the approach of checking schedulability. The approach of classifying delay bounds has complexity $O(L_C)$, where L_C represents the number of predefined delay classes. It was indicated in [FiKT97] that the value

of L_C can be much smaller than N_f (i.e., $L_C < N_f$). Thus, the approach of classifying delay bounds may have lower complexity than our algorithm. However, we know from Table 3-1 that it achieves lower bandwidth utilization than ours. In addition, as discussed in the remarks of Proposition 3.3, the computing operation of our algorithm is the simple subtraction. Both the checking schedulability approach and the classifying delay bounds approach need more complicated operation of multiplication. Table 3-2 shows that the peak rate reservation method has very low complexity and simple computation, however, Table 3-1 has shown that bandwidth utilization achieved by peak rate reservation approach is extremely low.

From the above three experiments and two comparison tables, we can see that our algorithm performs better than the previous algorithms in terms of both the bandwidth utilization and the computation time.

3.5 Chapter Summary and Remarks

In this chapter, we have presented a new admission control algorithm for real-time VBR traffic in the EDF-based RCS discipline network. The deterministic guarantee on end-to-end delay bound is adopted as the measure of quality-of-service (QoS). Unlike the previous studies, we concentrated on designing an efficient bandwidth reservation scheme, in which the service curve and the spare capacity curve are two important features. By providing the approaches to allocate the service curve and to update the spare capacity curve, we can support the deterministic guarantee on the end-to-end delay and achieve reasonable network utilization. We evaluate our algorithm in terms of both the network bandwidth utilization and the computational time needed to admit one connection. The experiments using the parameters derived from some real video traces show that our admission control algorithm is suitable to support real-time traffic and it can achieve reasonable network bandwidth utilization.

Chapter 4

Proposals on Lossless Traffic Shaping

4.1 Introduction

We design the lossless MLB shapers for VBR traffic. The shapers are designed for the network node model described in Fig. 2-2, in which each traffic flow is shaped at the shaping buffer before entering the multiplexer.

Two minimum MLB shaper designs are presented in this chapter. The first shaper design can guarantee the specified worst-case shaping delay bound, and the second shaper design can guarantee the specified worst-case shaping buffer occupancy. Furthermore, we investigate the node delay bound performance of the minimum MLB shaper, and show that the node delay bound of the minimum MLB shaper guaranteeing the specified shaping delay bound is not larger than the delay bound without traffic shaper.

The incoming traffic flow is modeled by using the MLB regulated traffic model that has been detailed in section 2.2.1. Let $\tilde{I}_j^*(t)$ denote the MLB traffic constraint function of the incoming flow j. Then, we have $\tilde{I}_j^*(t) = \min_{1 \le n \le \tilde{N}_j} (\tilde{\rho}_{j,n} t + \tilde{\sigma}_{j,n})$.

4.2 Lossless MLB Shaper Design

In this section, we present a few propositions on the MLB shaper designs for the traffic flow j.

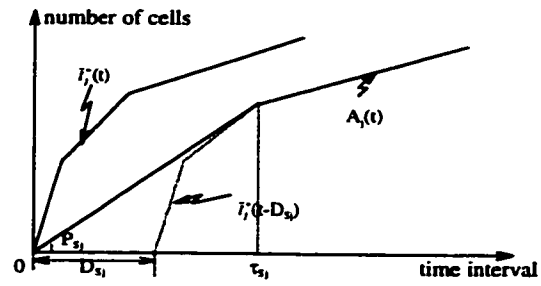


Fig. 4-1 A minimum MLB Shaper Guaranteeing the Shaping Delay Bound D_{sj}

Proposition 4.1: A minimum MLB shaper design for the shaping delay bound

Assume that the incoming traffic flow *j* at the shaper is MLB regulated with the traffic constraint function $\tilde{I}_j^*(t)= \min_{1\le n\le \tilde{N}_j} (\tilde{\rho}_{j,n}t + \tilde{\sigma}_{j,n})$, and that the shaping delay bound of traffic flow *j* is given as D_{sj}. The minimum MLB shaper for traffic flow *j* that satisfies the shaping delay bound D_{sj} can be constructed in the following steps. With reference to Fig. 4-1, we first shift $\tilde{I}_j^*(t)$ to the right for a distance of D_{sj}, and obtain function $\tilde{I}_j^*(t - D_{sj})$. Then, we can draw at least one line from the point (0, 0) to the function $\tilde{I}_j^*(t - D_{sj})$, and obtain the tangent point $(\tau_{sj}, \tilde{I}_j^*(\tau_{sj} - D_{sj}))$ with the smallest $\tau_{sj} > 0$ and $\tilde{I}_j^*(\tau_{sj} - D_{sj}) > 0$. Thus, the tangent line function with $0 \le t < \tau_{sj}$ and the function $\tilde{I}_j^*(t - D_{sj})$ with $t \ge \tau_{sj}$ construct the envelope function of the minimum MLB shaper. Let $A_j(t)$ denote the envelope function of this minimum shaper, and we have:

$$A_j(t) = \begin{cases} P_{Sj} \cdot t & \text{for } 0 \le t < \tau_{Sj} \\ \tilde{I}_j^*(t - D_{Sj}) & \text{for } t \ge \tau_{Sj} \end{cases}$$

where,

$$\tau_{sj} = D_{sj} + \min\{\tilde{\tau}_{j,n}: \tilde{\tau}_{j,n} > 0 \text{ and } \tilde{\rho}_{j,n} \le \tilde{I}_j^*(\tilde{\tau}_{j,n})/(\tilde{\tau}_{j,n} + D_{sj}) \le \tilde{\rho}_{j,n-1}\};$$

$P_{sj}=\frac{\tilde{I}_j^*(\tau_{Sj}-D_{Sj})}{\tau_{Sj}}$ is the peak rate of the MLB shaper;

$\tilde{\tau}_{j,n}$ is the X-axis value at the *n*th flex points in $\tilde{I}_j^*(t)$.

Proof: see Appendix B.1.

Please note that the value of D_{sj} in Proposition 4.1 should not be larger than $\tilde{I}_j^*(\tilde{\tau}_{j,N_j+1})/\tilde{\rho}_{j,N_j}-\tilde{\tau}_{j,N_j+1}$, because when $D_{sj} = \tilde{I}_j^*(\tilde{\tau}_{j,N_j+1})/\tilde{\rho}_{j,N_j}-\tilde{\tau}_{j,N_j+1}$, we say that the input traffic is fully shaped and the envelope function of the shaper is $A_j(t) = \tilde{\rho}_{j,\tilde{N}_j}\cdot t$. If $A_j(t) < \tilde{\rho}_{j,\tilde{N}_j}\cdot t$, the shaping delay bound would be infinity.

As an example, we can apply Proposition 4.1 to the DLB shaper design. For an incoming traffic flow *j* characterized by the DLB constraint function $\tilde{I}_j^*(t) = \min(\tilde{P}_j\cdot t, \tilde{\rho}_j\cdot t+\tilde{\sigma}_j)$, we construct its minimum DLB shaper by first shifting the function $\tilde{I}_j^*(t)$ to the right for a distance of D_{sj}. Then, we draw a tangent line from the point (0, 0) to the function $\tilde{I}_j^*(t - D_{sj})$, and obtain the tangent point $(\tau_{sj}, \tilde{I}_j^*(\tau_{sj} - D_{sj}))$ where $\tau_{sj} = D_{sj} + \tilde{\tau}_j$ and $\tilde{\tau}_j = \tilde{\sigma}_j / (\tilde{P}_j - \tilde{\rho}_j)$. Thus, we determine the parameters for the minimum DLB shaper: the long-term average rate $\gamma_j = \tilde{\rho}_j$, the maximum burst size $\beta_j = \tilde{\sigma}_j - \tilde{\rho}_j\cdot D_{sj}$, and the peak rate $P_{sj} = \tilde{P}_j\cdot\tilde{\tau}_j /(\tilde{\tau}_j + D_{sj})$. The envelope function of this minimum DLB shaper is $A_j(t) = \min(P_{sj}t, \gamma_j t + \beta_j)$.

•

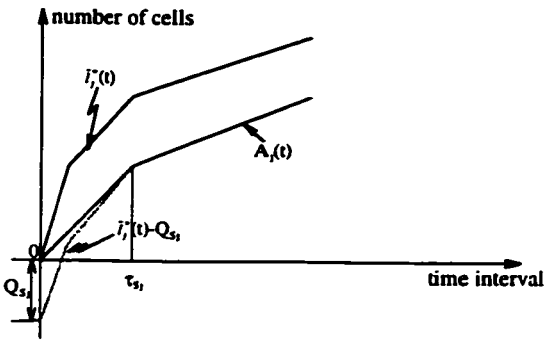


Fig. 4-2 A minimum MLB Shaper Guaranteeing the Worst-Case Shaping Buffer Occupancy Q_{sj}

Proposition 4.2: A minimum MLB shaper design for the worst-case buffer occupancy

Assume that the incoming traffic flow *j* at the shaper is MLB regulated with the traffic constraint function $\tilde{I}_j^*(t) = \min_{1 \le n \le \tilde{N}_j} (\tilde{\rho}_{j,n} t + \tilde{\sigma}_{j,n})$, and that the worst-case shaping buffer occupancy of connection *j* is given as Q_{Sj}. The minimum MLB shaper for traffic flow *j* that satisfies the worst-case shaping buffer occupancy Q_{Sj} can be constructed in the following steps. With reference to Fig. 4-2, we first shift $\tilde{I}_j^*(t)$ downwards for a distance of Q_{Sj}, and obtain function $y(t) = \tilde{I}_j^*(t) - Q_{Sj}$. Then, we can draw at least one line from the point (0, 0) to the function $y(t) = \tilde{I}_j^*(t) - Q_{Sj}$, and obtain the tangent point $(\tau'_{Sj}, \tilde{I}_j^*(\tau'_{Sj}) - Q_{Sj})$ with the smallest $\tau'_{Sj} > 0$ and $y(\tau'_{Sj}) > 0$. Thus, the tangent line function with $0 \le t < \tau'_{Sj}$ and the function $y(t) = \tilde{I}_j^*(t) - Q_{Sj}$ with $t \ge \tau'_{Sj}$ construct the envelope function of the minimum MLB shaper. Let $A_j(t)$ denote the envelope function of this minimum shaper, and we have:

$$A_j(t) = \begin{cases} P'_{Sj} \cdot t & \text{for } 0 \le t < \tau'_{Sj} \\ \tilde{I}_j^*(t) - Q_{Sj} & \text{for } t \ge \tau'_{Sj} \end{cases}$$

where,

$$\tau'_{Sj} = \min\{\tilde{\tau}_{j,n} : \tilde{\tau}_{j,n} > 0 \text{ and } \tilde{\rho}_{j,n} \le \frac{\tilde{I}_j^*(\tilde{\tau}_{j,n}) - Q_{Sj}}{\tilde{\tau}_{j,n}} \le \tilde{\rho}_{j,n-1}\}.$$

$P'_{Sj} = \dfrac{\tilde{I}_j^*(\tau'_{Sj}) - Q_{Sj}}{\tau'_{Sj}}$ is the peak rate of the MLB shaper.

Proof: see Appendix B.2.

Please note that the value of Q_{Sj} in Proposition 4.2 should not be larger than $\tilde{I}_j^*(\tilde{\tau}_{j,\tilde{N}_j+1}) - \tilde{\rho}_{j,\tilde{N}_j} \cdot \tilde{\tau}_{j,\tilde{N}_j+1}$, because when $Q_{Sj} = \tilde{I}_j^*(\tilde{\tau}_{j,\tilde{N}_j+1}) - \tilde{\rho}_{j,\tilde{N}_j} \cdot \tilde{\tau}_{j,\tilde{N}_j+1}$, we say that the input traffic is fully shaped and the envelope function of the shaper is $A_j(t) = \tilde{\rho}_{j,\tilde{N}_j} \cdot t$. If $A_j(t) < \tilde{\rho}_{j,\tilde{N}_j} \cdot t$, the worst-case shaping buffer occupancy would be infinity.

Similarly, we can apply Proposition 4.2 to the DLB shaper design. For an incoming traffic flow j characterized by the DLB constraint function $\tilde{I}_j^*(t) = \min(\tilde{P}_j \cdot t, \tilde{\rho}_j \cdot t + \tilde{\sigma}_j)$, we construct its minimum DLB shaper by first shifting the function $\tilde{I}_j^*(t)$ downwards for a distance of Q_{Sj}. Then, we draw a tangent line from point (0, 0) to the function $y(t) = \tilde{I}_j^*(t) - Q_{Sj}$, and obtain the tangent point $(\tau'_{Sj}, y(\tau'_{Sj}))$ where $\tau'_{Sj} = \tilde{\tau}_j$ and $\tilde{\tau}_j = \tilde{\sigma}_j / (\tilde{P}_j - \tilde{\rho}_j)$. Thus, we determine the parameters for the minimum DLB shaper: the long-term average rate $\gamma_j = \tilde{\rho}_j$, the maximum burst size $\beta_j = \tilde{\sigma}_j - Q_{Sj}$, and the peak rate $P'_{Sj} = \tilde{P}_j - Q_{Sj} / \tilde{\tau}_j$.

The use of traffic shaper can reduce the traffic distortion and the resource requirements, but it may introduce extra delay for traffic flows because there can be packets waiting in the shaper even though the output link is idle [GGPS96]. In the following proposition, we show that using traffic shaper does not always increase the node delay bound.

Proposition 4.3: ***The worst-case node delay performance of the minimum MLB shaper guaranteeing the shaping delay bound***

Assume that the incoming traffic flow j at the shaper is MLB regulated with the traffic constraint function $\tilde{I}_j^*(t)= \min_{1\le n\le \tilde{N}_j} (\tilde{\rho}_{j,n}t + \tilde{\sigma}_{j,n})$. Let c_j be the bandwidth allocated to connection j. If the shaper design presented in Proposition 4.1 is used, then the worst-case node delay incurred on connection j remains the same as that when no traffic shaper is used.

Proof: see Appendix B.3.

Both the sufficient and necessary conditions can be shown for the DLB shaper. That is, *given that certain bandwidth c_j has been allocated to connection j, the node delay bound when a DLB shaper is used remains the same as that when no traffic shaper is used if and only if the shaper is designed according to Proposition 4.1.* The sufficient condition is the direct result of Proposition 4.3. The necessary condition can be proved by following contradiction.

According to the construction of $A_j(t)$ in Proposition 4.1, it must be true that $A_j(\tau_{sj}) = \tilde{I}_j^*(\tilde{\tau}_j)$ and the node delay is bounded by $D_j = \tilde{I}_j^*(\tilde{\tau}_j) / c_j - \tilde{\tau}_j$. If the shaper is not designed according to Proposition 4.1, then $A_j(\tau_{sj}) > \tilde{I}_j^*(\tilde{\tau}_j)$ or $A_j(\tau_{sj}) < \tilde{I}_j^*(\tilde{\tau}_j)$. If $A_j(\tau_{sj}) < \tilde{I}_j^*(\tilde{\tau}_j)$, the node delay is bounded by

$$D^* = [\, A_j^{-1}(\tilde{I}_j^*(\tilde{\tau}_j)) - \tilde{\tau}_j] + [\, A_j(\tau_{sj}) / c_j - \tau_{sj}\,] \quad .$$

The first part is the worst-case shaping delay and the second part is the worst-case multiplexing delay. Because $c_j \ge \rho_j$,

$$D^* \ge [A_j^{-1}(\tilde{I}_j^*(\tilde{\tau}_j)) - \tilde{\tau}_j] + [\tilde{I}_j^*(\tilde{\tau}_j) / c_j - A_j^{-1}(\tilde{I}_j^*(\tilde{\tau}_j))]$$

$$= \tilde{I}_j^*(\tilde{\tau}_j) / c_j - \tilde{\tau}_j = D_j \quad .$$

In the above, $A_j^{-1}(y)$ is the inverse function of $A_j(t)= \min_{t\geq 0}(P_{sj}t, \gamma_j t + \beta_j)$, i.e., $A_j^{-1}(y) = \max_{y\geq 0}(y/P_{sj}, (y - \beta_j)/\gamma_j)$.

If $A_j(\tau_{sj}) > \tilde{I}_j^*(\tilde{\tau}_j)$, the node delay bound is

$$D^* = [A_j^{-1}(\tilde{I}_j^*(\tilde{\tau}_j)) - \tilde{\tau}_j] + [A_j(\tau_{sj}) / c_j - \tau_{sj}] \quad .$$

Because $P_{sj} \geq c_j$,

$$D^* \geq [A_j^{-1}(\tilde{I}_j^*(\tilde{\tau}_j)) - \tilde{\tau}_j] + [\tilde{I}_j^*(\tilde{\tau}_j)/c_j - A_j^{-1}(\tilde{I}_j^*(\tilde{\tau}_j))]$$

$$= \tilde{I}_j^*(\tilde{\tau}_j) / c_j - \tilde{\tau}_j = D_j \quad .$$

Thus, the necessary condition is proved.

4.3 Performance Evaluation

Performance evaluation is presented in this section to evaluate the shaper designs of the above section. First, we obtain, for a typical video trace connection, the designing parameters of the two types of minimum traffic shaper that can guarantee the specified shaping delay bound or the specified shaping buffer occupancy bound. Second, we have proved and demonstrated that these obtained shapers are the minimum shapers that can satisfy the given shaping delay bound or the worst-case shaping buffer occupancy. That is, they have the minimum envelope function among all those shapers that can satisfy the given shaping delay bound or shaping buffer occupancy bound.

The video trace used in this experiment is characterized by the envelope function $\tilde{I}^*(t) = \min_{1\leq n\leq 4} (\tilde{\rho}_n \cdot t + \tilde{\sigma}_n)$. The slope values of its four line segments are $\tilde{\rho}_1$ = 4000 cells/sec, $\tilde{\rho}_2$ = 667 cells/sec, $\tilde{\rho}_3$ = 600 cells/sec, and $\tilde{\rho}_4$ = 500 cells/sec. The y-intercept values of its four line segments are $\tilde{\sigma}_1$ = 0, $\tilde{\sigma}_2$ = 267 cells, $\tilde{\sigma}_3$ = 533 cells, and $\tilde{\sigma}_4$ = 1133 cells.

video trace	$\rho 1 = 4000$	$\sigma 1 = 0$	$\rho 2 = 667$	$\sigma 2 = 267$	$\rho 3 = 600$	$\sigma 3 = 533$	$\rho 4 = 500$	$\sigma 4 = 1133$	bdwd alloc. = 2000 cells/sec
Ds =0.02 sec	$\gamma 1 = 3200$	$\beta 1 = 0$	$\gamma 2 = 667$	$\beta 2 = 253$	$\gamma 3 = 600$	$\beta 3 = 521$	$\gamma 4 = 500$	$\beta 4 = 1123$	Nddwosh = Nddwsh = 0.08 second
Ds =0.05 sec	$\gamma 1 = 2461$	$\beta 1 = 0$	$\gamma 2 = 667$	$\beta 2 = 233$	$\gamma 3 = 600$	$\beta 3 = 503$	$\gamma 4 = 500$	$\beta 4 = 1108$	Nddwosh = Nddwsh = 0.08 second
Ds = 0.5 sec	$\gamma 1 = 652$	$\beta 1 = 0$	$\gamma 2 = 600$	$\beta 2 = 233$	$\gamma 3 = 500$	$\beta 3 = 883$			Nddwosh = Nddwsh = 0.08 sec
Qs = 40 cells	$\gamma 1 = 3500$	$\beta 1 = 0$	$\gamma 2 = 667$	$\beta 2 = 227$	$\gamma 3 = 600$	$\beta 3 = 493$	$\gamma 4 = 500$	$\beta 4 = 1093$	Bufwosh = Bufwsh = 160 cells
Qs =100 cells	$\gamma 1 = 2750$	$\beta 1 = 0$	$\gamma 2 = 667$	$\beta 2 = 167$	$\gamma 3 = 600$	$\beta 3 = 433$	$\gamma 4 = 500$	$\beta 4 = 1033$	Bufwosh = Bufwsh = 160 cells
Qs =400 cells	$\gamma 1 = 633$	$\beta 1 = 0$	$\gamma 2 = 600$	$\beta 2 = 133$	$\gamma 3 = 500$	$\beta 3 = 733$			Bufwosh = Bufwsh = 160 cells

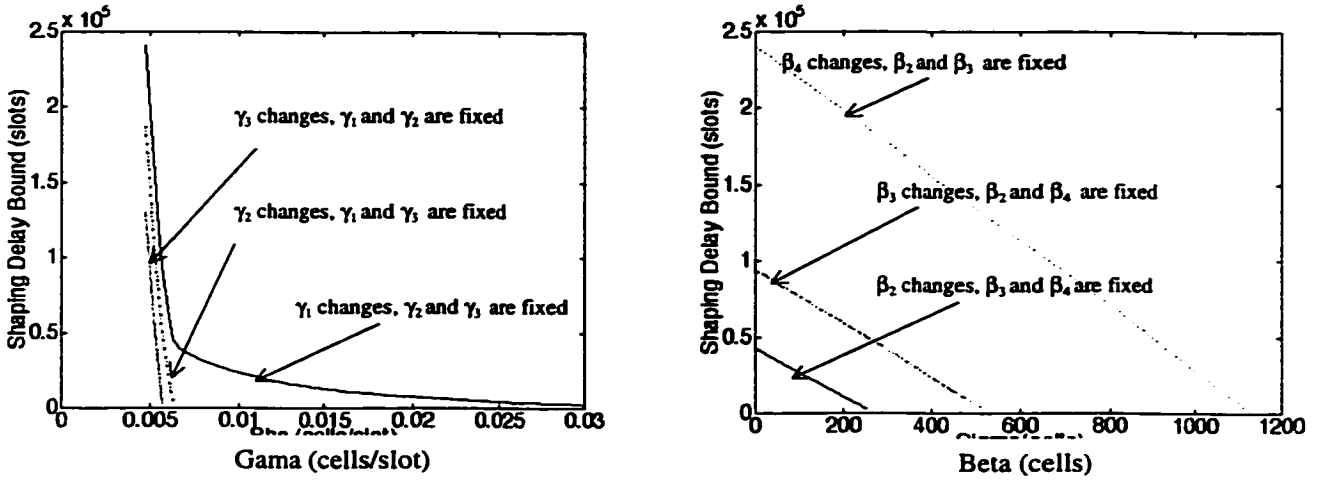
Table 4-1 Traffic Shaper Designs for Various D_S and Q_S Values

For this traffic trace, we construct three minimum shapers that can guarantee three shaping delay bounds 0.02, 0.05 and 0.5 second, respectively. Their designing parameters can be found in Table 4-1. For instance, the minimum shaper satisfying shaping delay bound 0.02 second has the parameters: γ_1 = 3200 cells/sec, β_1 = 0, γ_2 = 667 cells/sec, β_2 = 253 cells, γ_3 = 600 cells/sec, β_3 = 521 cells, γ_4 = 500 cells/sec, and β_4 = 1123 cells. The envelope function of this minimum shaper is $A(t) = \min_{1 \le k \le 4} \{\gamma_k t + \beta_k\}$.

From the table, we also see that when the bandwidth of 2000 cells/sec is allocated to this video traffic flow, the worst-case node delay is 0.08 second when no shaper is used or when the above minimum shapers are used.

Similarly, we have constructed for this video trace three minimum shapers that guarantee three worst-case shaping buffer occupancies 40, 100, and 400 cells, respectively. Their designing parameters can also be found in Table 4-1. For instance, the minimum shaper satisfying the worst-case shaping buffer occupancy 400 cells has the parameters: γ_1 = 633 cells/sec, β_1 = 0, γ_2 = 600 cells/sec, β_2 = 133 cells, γ_3 = 500 cells/sec, and β_3 = 733 cells. The envelope function of this minimum shaper is $A(t) = \min_{1 \le k \le 3} \{\gamma_k t + \beta_k\}$. We also see from the table that, with the bandwidth of 2000 cells/sec allocated to this video traffic flow, the worst-case node buffer occupancy is 160 cells when no shaper is used or when the above obtained minimum shapers are used.

We demonstrate next that the shaper designs presented in Table 4-1 are the minimum shapers that can satisfy the specified shaping delay bound or the specified shaping buffer occupancy bound. We first construct shapers with smaller envelopes, and then illustrate that these smaller shapers may cause larger shaping delay bound or larger shaping buffer occupancy than those in Table 4-1.

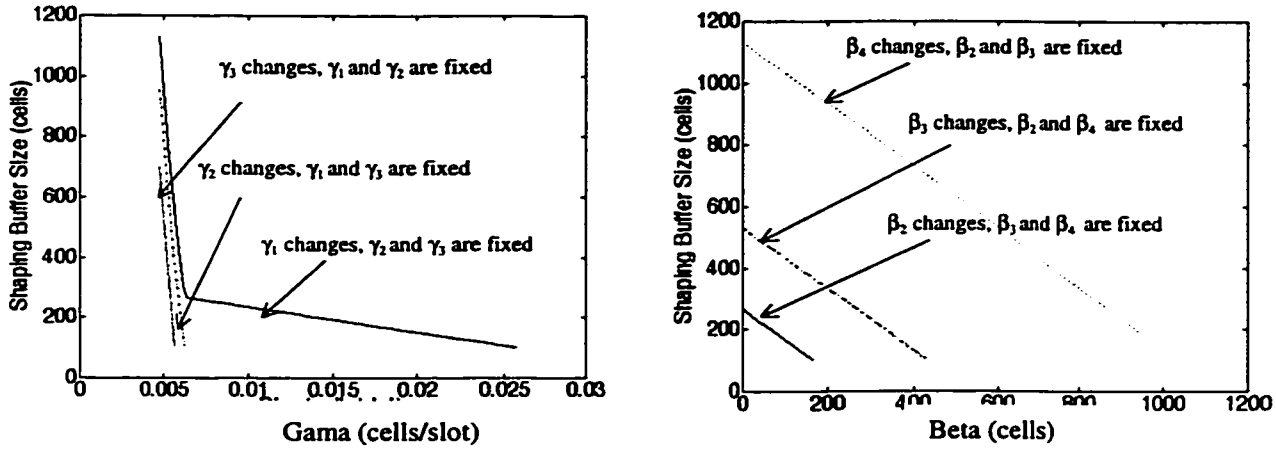


(a) With smaller γ values

(b) With smaller β values

Fig. 4-3 Shaping Delay Bound with Smaller Shaper Envelope

It can be seen from Table 4-1 that the shaping delay bound 0.02 second can be satisfied by using the shaper with parameters γ_1 =3200 cells, $\beta_1 = 0$ cells, $\gamma_2 = 667$ cells/sec, $\beta_2 = 253$ cells, $\gamma_3 =$ 600 cells/sec, $\beta_3 = 521$ cells, $\gamma_4 = 500$ cells/sec, and $\beta_4 = 1123$ cells. Here, we construct traffic shapers with smaller envelopes by reducing the values of γ's or β's. The shaping delay bound with smaller γ values are shown in Fig. 4-3 (a). It can be seen that when γ_1, γ_2, or γ_3, decreases from those values in Table 4-1, the shaping delay bound becomes larger than 0.02 second. Please note that γ_4 cannot decrease below the traffic trace's sustainable cell rate of 500 cells/sec; otherwise the shaping delay bound would be infinity. The shaping delay bounds with smaller β values are shown in Fig. 4-3 (b). We can see that shaping delay bound becomes larger than 0.02 second when the values of β_2, β_3, or β_4 decreases from those values in Table 4-1. Note that β_1 cannot decrease below 0 due to the property of MLB shapers.



(a) With smaller γ values

(b) With smaller β values

Fig. 4-4 Worst-case shaping buffer occupancy with smaller shaper envelope

Similarly, it can be found in Table 4-1 the worst-case shaping buffer occupancy 100 cells can be satisfied by using the shaper with parameters γ_1 =2750 cells, β_1 = 0 cells, γ_2 = 667 cells/sec, β_2 = 167 cells, γ_3 = 600 cells/sec, β_3 = 433 cells, γ_4 = 500 cells/sec, and β_4 = 1033 cells. Here, we construct traffic shapers with smaller envelopes by reducing the values of γ's or β's. The shaping buffer occupancy bounds with smaller γ values are shown in Fig. 4-4 (a). We can see that the worst-case shaping buffer occupancy becomes larger than 100 cells when γ_1, γ_2, or γ_3, decreases from those values in Table 4-1. Please note that γ_4 cannot decrease below the traffic trace's sustainable cell rate 500 cells/sec; otherwise the worst-case buffer occupancy would be infinity. The worst-case shaping buffer occupancies with smaller β values are shown in Fig. 4-4 (b). It can be seen that the shaping buffer occupancy becomes larger than 100 cells when the values of β_2, β_3, or β_4 decreases below those values in Table 4-1. Due to the property of MLB shaper, the value of β_1 cannot decrease below zero.

4.4 Chapter Summary and Remarks

In this chapter, we have presented two minimum MLB shaper designs that can satisfy the specified worst-case shaping buffer occupancy or the specified shaping delay bound. Furthermore, we show that the node delay bound of the minimum MLB shaper guaranteeing the specified shaping delay bound is not larger than the delay bound without traffic shaper. Performance evaluation has been conducted to evaluate the presented lossless MLB shapers.

Chapter 5

Statistical Traffic Shaping

5.1 Introduction

In order to improve the network resource utilization, we propose in this chapter to allow a small loss probability at the traffic shaper as well as at the multiplexer. We call our traffic shaping scheme *statistical traffic shaping* here. The statistical traffic shaping is proposed for the network model of Fig. 2-2 where each traffic flow is shaped at the shaping buffer before entering the multiplexer.

Analysis is carried out for both cases of homogeneous traffic class and heterogeneous traffic classes in order to investigate the condition under which statistical shaping is more beneficial for improving resource utilization than lossless shaping. Our performance evaluation shows that a higher level of network resource utilization can be achieved when statistical traffic shaping is used than that when lossless shaping is used, and than that when no cell loss is guaranteed at the node.

With reference to Fig. 2-4 and section 2.2.2, the VBR traffic flows offered to a network node are assumed to be extremal, on-off and periodic processes with independent and uniformly distributed random phases, i.e., the incoming rate process of connection j can be illustrated as $\Omega_j(t)$ in Fig. 2-4. We also make the same assumption as in section 2.3, that is, the incoming traffic flows can be grouped into J classes, and there are K_j connections in class j. All connections in class j are associated with the same set of DLB parameters ($\tilde{P}_j$, $\tilde{\sigma}_j$, $\tilde{\rho}_j$), but they

have different phases. That is, $\Omega_{ji}(t) = \Omega_j(t + \theta_{ji})$, where $1 \le i \le K_j$, $1 \le j \le J$, θ_{ji} is an independent random variable uniformly distributed in $[0, T_j]$, and T_j is the period of process $\Omega_j(t)$.

5.2 Allowing a small loss probability at the traffic shaper

We propose to allow a very small loss probability in our statistical traffic shaping scheme. We use upper bounding technique to estimate the loss probability at the shaper, and investigate the condition of achieving the statistical multiplexing gain at the shaper.

5.2.1 Upper Bound of Cell Loss Probability at the Traffic Shaper

In this section, we borrow the analysis technique in [ElMW95], and apply it to study the loss probability at the traffic shaper. When considering cell loss probability, we first observe that the data buffer is the only physical resource at the DLB traffic shaper. The token buffer is essentially a virtual device to count the number of departures in one burst. With reference to Fig. 2-6 (b) and section 2.3, we let $q_{ji}(t)$ be the instantaneous shaping buffer demand from the ith connection of class j at time t, and B_S be the total shaping buffer capacity at the node. Then at this single resource system, the cell loss is incurred when the total instantaneous demand for the data buffer $\sum_{j=1}^{J}\sum_{i=1}^{K_j} q_{ji}(t)$ exceeds the data buffer capacity B_S. Let P_{LOSS} be the loss probability at the shaper, then P_{LOSS} measures the fraction of time that the total demand for the data buffer from all the traffic flows exceeds the data buffer occupancy. That is, $P_{LOSS} = Pr\left\{ \sum_{j=1}^{J}\sum_{i=1}^{K_j} q_{ji}(t) > B_S \right\}$.

Two assumptions are required for deriving P_{LOSS} the loss probability at the shaper:

(1) $\sum_{j=1}^{J} K_j E[q_{ji}] \le B_S$, since otherwise the system is unstable.

(2) $\lim_{\theta \to \infty} \sum_{j=1}^{J} K_j \frac{G_j'(\theta)}{G_j(\theta)} > B_S$, since otherwise there is no loss, where $G_j(\theta) = E[exp(\theta \cdot q_{ji})]$ and parameterθ is left unspecified [Buck90, ElMW95].

In order to simplify the computation of P_{LOSS}, we conservatively bound the process $q_j(t)$ in Fig. 2-6 (b) by an on-off process. We assign a peak value of b_{Sj} for the on period with probability ω_{R_j} and the value 0 for the off period with probability (1 - ω_{R_j}). Then, using the large deviation theory [Buck90, MiMo95], we obtain the Chernoff's bound on P_{LOSS}:

$$\log(P_{LOSS}) \leq -\Phi(\theta^*)$$

where,

$$\Phi(\theta) = \theta \cdot B_S - \sum_{j=1}^{J} K_j \cdot \log[1 - \omega_{R_j} + \omega_{R_j} \exp(\theta \cdot b_{Sj})] \quad , \tag{5-1}$$

$$\Phi(\theta^*) = \sup_{\theta \geq 0} \{\Phi(\theta)\} \quad ,$$

and θ^* can be obtained by solving the equation $\Phi'(\theta) = 0$, i.e.,

$$\sum_{j=1}^{J} \frac{K_j \omega_{R_j} b_{Sj} \exp(\theta \cdot b_{Sj})}{1 - \omega_{R_j} + \omega_{R_j} \exp(\theta \cdot b_{Sj})} = B_S \quad . \tag{5-2}$$

Therefore $\exp(-\Phi(\theta^*))$ upper bounds the probability that the total instantaneous requirement for the shaping buffer exceeds B_S.

5.2.2 Statistical Multiplexing Gain of the Traffic Shaper

We consider here the case of single traffic class (i.e., J = 1), and we analyze the condition of achieving the statistical multiplexing gain at the traffic shaper.

Using J = 1 in equations (5-1) and (5-2), we obtain

$$\Phi(\theta^*) = \theta^* \cdot B_S - K \cdot \log(1 - \omega_R + \omega_R \cdot \exp(\theta^* \cdot b_S)) \quad . \tag{5-3}$$

where,

$$\theta^* = \frac{1}{b_S}\log\frac{B_S(1-\omega_R)}{\omega_R(Kb_S - B_S)} \quad ; \tag{5-4}$$

K is the number of traffic flows shaped from ($\tilde{P}$, $\tilde{\sigma}$, $\tilde{\rho}$) to (P_S, β, γ).

It is easy to show from eqn. (5-4) that $\theta^* \geq 0$. From eqn. (5-3), it can be proved that $\frac{\partial\Phi(\theta^*)}{\partial K} \leq 0$. Using a derivation similar to [ElMW95], the function $\Phi(\theta^*)$ can be shown to have the following features: (i) $\lim_{K\to\frac{B_S}{b_S}} \frac{\partial\Phi(\theta^*)}{\partial K} = -\infty$, (ii) $\lim_{K\to\frac{B_S}{b_S}} \Phi(\theta^*) = \frac{B_S}{b_S}\log(\frac{1}{\omega_R})$, and (iii) $\lim_{K\to\frac{B_S}{\omega_R b_S}} \Phi(\theta^*) = 0$. Therefore, with fixed values of B_S, ω_R, and b_S, $\Phi(\theta^*)$ is strictly decreasing with increasing K when K is within the interval [B_S/b_S, $B_S/(\omega_R b_S)$]. We can also show easily that $\frac{\partial}{\partial K}(\frac{\partial\Phi(\theta^*)}{\partial K}) > 0$ for $\frac{B_S}{b_S} \leq K \leq \frac{B_S}{\omega_R b_S}$. This means, the slope of $\Phi(\theta^*)$ becomes smoother with increasing K when K is within the interval $[\frac{B_S}{b_S}, \frac{B_S}{\omega_R b_S}]$. Hence, $\Phi(\theta^*)$ can be illustrated using Fig. 5-1.

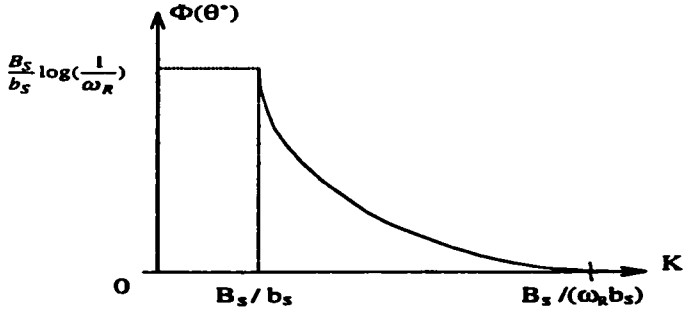


Fig. 5-1 Illustration of $\Phi(\theta^*)$ as a Function of K

Based on the results of analysis in section 5.2.1, we have $\log(1 / P_{LOSS}) \geq \Phi(\theta^*)$. Let L_S be the QoS requirement of loss probability at the shaper, then we must have $P_{LOSS} \leq L_S$. When $L_S = 0$, we must have $P_{LOSS} = 0$. In this case, $K = B_S / b_S$. When a small loss probability $L_S > 0$ is allowed at the shaper, we are interested in finding the maximum value of K for which $P_{Loss} \leq L_S$. The maximum value of K is obtained by the solution of $\Phi(\theta^*) = \log(1/L_S)$. From Fig. 5-1, we can observe that there is no solution for $\Phi(\theta^*) = \log(1 / L_S)$ if $\log(1 / L_S) \geq (B_S / b_S) \cdot \log(1 / \omega_R)$. In this case, $K = B_S / b_S$ and $P_{Loss} = 0$. At the same time, we observe that the maximum value of K is larger than B_S / b_S if $\log(1/L_S) < (B_S/b_S)\log(1/\omega_R)$, i.e., the statistical multiplexing gain is achieved when $B_S > b_S \log(1/L_S) / \log(1/\omega_R)$.

Fig. 5-1 shows that $\Phi(\theta^*)$ is decreasing with respect to the increasing K when $\Phi(\theta^*) = \log(1 / L_S) < (B_S / b_S) \cdot \log(1 / \omega_R)$. This means, more connections can be shaped from $(\tilde{P}, \tilde{\sigma}, \tilde{\rho})$ to (P_S, β, γ) when larger loss probability is allowed at the shaper. In another word, larger statistical multiplexing gain can be obtained at the shaper with larger loss probability allowed. In this chapter, we fix the number of input connections at the shaper as K_{S0}, and fix the DLB traffic parameters $(\tilde{P}, \tilde{\sigma}, \tilde{\rho})$ of each input stream. Thus when larger statistical multiplexing gain is achieved, K_{S0} traffic flows can be shaped into a less bursty pattern (e.g., with a smaller P_S). When statistical multiplexing gain is achieved at the shaper, we are interested in obtaining P_S the peak rate of the shaped traffic flow. There is no close-form solution for P_S, but it can be obtained by solving the equation $\log(1 / L_S) = \Phi(\theta^*)|_{K=K_{S0}, \omega_R=\gamma/P_S, b_S=(\tilde{P}-P_S)\tilde{\tau}}$, iteratively.

5.3 Critical Value Designs

Our original goal of allowing a small loss probability at the shaper is to be able to shape the input flows into a less bursty pattern and thereby to improve the network resource utilization when less bursty traffic flows are multiplexed. However, cell loss probability usually affects Quality of Service. As an example, let L denote the QoS requirement of the loss probability at a network node, and let L_M and L_S represent the QoS requirement of the loss probability at the multiplexer and at the shaper, respectively. Then, we have

$$\begin{aligned} L &= 1 - (1 - L_S)(1 - L_M) \\ &= L_M + L_S - L_M \cdot L_S \\ &\cong L_M + L_S \qquad \text{(for very small } L_M \text{ and } L_S\text{)} \ . \end{aligned}$$

The above relationship of $L \cong L_S + L_M$ indicates that $L_M \cong L - L_S$. Thus, for a given loss requirement L, the value of L_M when statistical shaping is used ($L_S > 0$) must be smaller than that when lossless shaping is used ($L_S = 0$). Hence, although statistical shaping may be able to achieve statistical multiplexing gain at the shaper, it may cause smaller statistical multiplexing gain achievable at the multiplexer due to smaller loss probability allowed at the multiplexer, and therefore it may result in lower resource utilization at the node. Hence, we shall investigate the condition under which statistical shaping is more beneficial for improving the resource utilization than lossless shaping. The following sections 5.3.1 and 5.3.2 discuss the critical value design in both cases of single traffic class and multiple traffic classes. The critical value design is for the shaping buffer capacity. The critical value of the shaping buffer capacity is defined as the threshold that statistical shaping is more beneficial than lossless shaping in terms of the

connection carrying capability if and only if the shaping buffer capacity is on one side of the critical value.

5.3.1 Homogeneous Traffic Class

In this case, all the incoming streams entering the node have the same traffic parameters ($\tilde{P}$, $\tilde{\sigma}$, $\tilde{\rho}$), and their rate-controllers also have the same parameters (P_S, β, γ), which means that each stream entering the multiplexer has the same traffic parameters (P_S, β, γ). Let K be the maximum number of admissible connections at the multiplexer, C be the output link capacity, e_0 be the bandwidth allocation of each connection, and γ be the long-term average rate of each input traffic flow at the multiplexer. Then, the Chernoff's bound on the loss probability L_M at the multiplexer can be shown [ElMW95] to be $\log(L_M) \le -F(s^*)$, where

$$F(s^*) = \frac{C}{e_0}\log(\frac{C}{K\gamma}) - (K - \frac{C}{e_0})\log(\frac{K(e_0-\gamma)}{Ke_0 - C}) \quad . \tag{5-5}$$

Furthermore, $F(s^*)$ has the following features: (i) $\frac{\partial F(s^*)}{\partial K} = \log(\frac{Ke_0 - C}{K(e_0-\gamma)}) < 0$; (ii) $\lim_{K \to C/e_0} \frac{\partial F(s^*)}{\partial K} = -\infty$; (iii) $\lim_{K \to C/e_0} F(s^*) = \frac{C}{e_0}\log(\frac{e_0}{\gamma}) \stackrel{def}{=} f(e_0)$, and (iv) $F(s^*)|_{K=C/\gamma} = 0$.

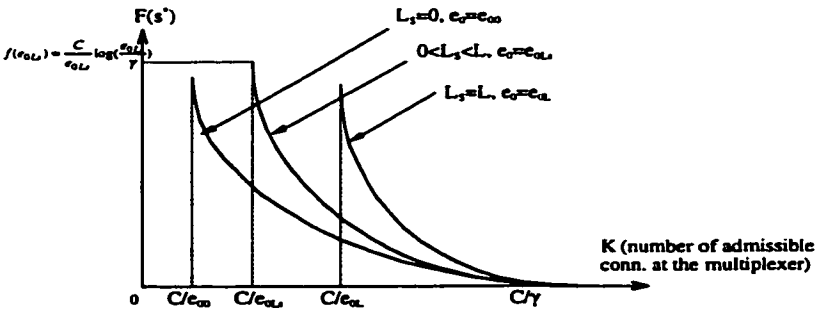


Fig. 5-2 $F(s^*)$ as a Function of the Number of Admissible Connections

Our starting point is eqn. (2-2). We can show that $\frac{\partial e_0}{\partial P_S} \geq 0$ and $\frac{\partial e_0}{\partial \beta} \geq 0$. This means that the bandwidth allocation e_0 decreases with decreasing P_S or decreasing β. Thus, less bandwidth needs to be allocated to the input connection when it has been shaped into a less bursty traffic pattern. Let P_{S0}, P_{SLs}, P_{SL} denote the peak rate of the shaped traffic flow when lossless shaping is used, when a small loss probability L_S is allowed at the shaper with $0 < L_S < L$, and when a loss probability L is allowed at the shaper, respectively. Then, e_{00}, the bandwidth allocation of each connection when lossless shaping is used, can be obtained by using eqn. (2-2) with $P_S = P_{S0}$. Similarly, e_{0Ls} and e_{0L}, the bandwidth allocation of each connection when loss probability L_S and L are allowed at the shaper, can be obtained by using eqn. (2-2) with $P_S = P_{SLs}$ and $P_S = P_{SL}$, respectively. From the discussion on Fig. 5-1, we know that $P_{SL} \leq P_{SLs} \leq P_{S0}$. Thus $e_{0L} \leq e_{0Ls} \leq e_{00}$. We also showed from eqn. (5-5) that $\frac{\partial}{\partial e_0}(\frac{\partial F(s^*)}{\partial K}) > 0$ for $K > C / \gamma$, which means the slope of the curve $F(s^*)$ will become steeper when the traffic is shaped in a larger degree. Thus, the $F(s^*)$ curves with various e_0 values can be illustrated using Fig. 5-2.

Based on the Chernoff's bound estimation of the loss probability at the multiplexer, the maximum number of admissible connections at the multiplexer when $L_S = 0$ can be obtained by solving $\log(1/L) = F(s^*)|_{e0=e00}$. When statistical shaping is used, the maximum number of admissible connections is $\max_{0<L_S \leq L} \left\{ K: \log(\frac{1}{L - L_S}) = F(s^*)|_{e_0 = e_{0Ls}} \right\}$.

Critical Value Design of Shaping Buffer Capacity B_S

Let $K(L_S=0, L_M=L)$ and $K(L_S>0, L_M=L-L_S)$ denote respectively the maximum number of admissible connections when lossless shaping is used and that when statistical shaping is used.

We define the critical value of the shaping buffer capacity to be the threshold that $K(L_S>0, L_M=L-L_S) > K(L_S=0,L_M=L)$ when the shaping buffer capacity is on one side of the threshold, and that $K(L_S>0,L_M=L-L_S) \leq K(L_S=0,L_M=L)$ when the shaping buffer capacity is on the other side of the threshold. That is, statistical shaping is more beneficial than lossless shaping in terms of the connection carrying capability if and only if the shaping buffer capacity is on one side of its critical value.

From eqn. (2-2), we know that the valid duration of e_0 is when $e_0 \geq \gamma$. Clearly, if B_S is so large that we have $e_{00} = \gamma$, then it must be true that $e_{0L_S} = e_{00} = \gamma$ and $f(e_{0L_S}) = f(e_{00}) = 0$ for $0 < L_S \leq L$. Hence, $K(L_S=0,L_M=L) = K(L_S>0,L_M=L-L_S) = C/\gamma$. That is, when B_S increases beyond a threshold value B_{Scrh}, statistical shaping will not be more beneficial for admitting more connections than lossless shaping. From section 2.3, we can easily derive two equations $\beta = \tilde{P} \cdot \tilde{\tau}(1-\gamma / P_{S0})$ and $K_{S0} \cdot (\tilde{P} - P_{S0}) \cdot \tilde{\tau} = B_S$. Using these two equations and eqn. (2-2), we obtain:

$$B_{Scrh} = \frac{K_{S0} \cdot \tilde{P}\tilde{\tau} \cdot [\tilde{P}\tilde{\tau} - \gamma\tilde{\tau} - \gamma(B_M / C)]}{\tilde{P}\tilde{\tau} - \gamma(B_M / C)} \quad . \qquad (5\text{-}6)$$

On the other hand, when $B_S \leq b_S \cdot \log(1/L)/\log(1/\omega_R)$, we must have $B_S \leq b_S \cdot \log(1/L_S)/\log(1/\omega_R)$. Thus, if B_S decreases beyond $b_S \cdot \log(1/L)/\log(1/\omega_R)$, there is no statistical multiplexing gain at the shaper. In this case, when the number of incoming connections is fixed as K_{S0}, we have $P_{S0} = P_{SL_S}$, and thus $e_{00} = e_{0L_S}$. From Fig. 5-1, if B_S is larger than but very close to $b_S \cdot \log(1/L)/\log(1/\omega_R)$, a very small statistical multiplexing gain could be obtained at the shaper, which means the values of P_{S0} and P_{SL} are close to each other. Then, the values of e_{00} and e_{0L} are close to each other, and thereby those $F(s^*)$ curves in Fig. 5-2 are close to each other. If the

maximum horizontal distance between two curves $F(s^*)|_{L_S=0}$ and $F(s^*)|_{L_S=L}$ is less than 1, the number of admissible connections when $L_S>0$ can *at most* be equal to that when $L_S=0$. Therefore, the other critical value of B_S must exist, i.e., when B_S decreases beyond certain value B_{Scrl}, $K(L_S=0,L_M=L) \geq K(L_S>0,L_M=L-L_S)$. Please note that B_{Scrl} can be obtained by using the procedure *find_*B_{Scrl} in Appendix C. 1.

Summarizing the above discussion, we present the following proposition as a criterion to choose the value of shaping buffer capacity.

Proposition 5.1: Assume that the values of B_M and C are fixed, then

(1) If $B_S \geq B_{Scrh}$, $K(L_S=0,L_M=L) = K(L_S>0,L_M=L-L_S)$;

(2) If $B_{Scrl} < B_S < B_{Scrh}$, $K(L_S=0,L_M=L) < K(L_S>0,L_M=L-L_S)$;

(3) If $B_S \leq B_{Scrl}$, $K(L_S=0,L_M=L) \geq K(L_S>0,L_M=L-L_S)$.

Hence, when B_S is within the interval (B_{Scrl},B_{Scrh}), statistical shaping is more beneficial than lossless shaping in terms of the connection carrying capability.

5.3.2 Heterogeneous Traffic Classes

We study the benefit of using statistical shaping in the heterogeneous environment. In the heterogeneous environment, the shaping buffer of size B_S is shared by multiple traffic classes. The burstiness of the shaped traffic must be tightly related to the shaping buffer allocation scheme. Since designing the shaping buffer allocation scheme is not the goal of this work, we discuss two fundamental buffer allocation schemes at the shaping buffer: complete-dividing and complete-sharing. We focus on investigating how the analysis results in section 5.3.1 for the

single traffic class can be extended to the case of multiple traffic classes under these two fundamental buffer allocation schemes.

A. Complete-dividing shaping buffer

When the shaping buffer of size B_S is completely divided among J traffic classes with $J>1$, we have $\sum_{j=1}^{J} B_{S,j} = B_S$ where $B_{S,j}$ represents the shaping buffer space reserved for the class-j traffic flows. In this case, the results in section 5.3.1 can be used directly to choose an appropriate value of B_S. According to Proposition 5.1, with $B_S \geq \sum_{j=1}^{J} B_{Scrh,j}$, statistical shaping does not help achieve higher connection carrying capability than lossless shaping. Thus, in order to achieve higher connection carrying capability, we should have B_S value satisfying $\sum_{j=1}^{J} B_{Scrl,j} < B_S < \sum_{j=1}^{J} B_{Scrh,j}$, and have the $B_{S,j}$ value satisfying $B_{Scrl,j} < B_{S,j} < B_{Scrh,j}$.

B. Complete-sharing shaping buffer

When the shaping buffer is completely shared among J classes of traffic flows with $J>1$, there is no portion of buffer reserved for any particular traffic class. In this case, one naturally thinks about the competition among traffic classes for the finite buffer space, and possible monopoly of the shaping buffer by certain traffic class with heavy and bursty load. However, the buffer resource competition is not the concern here, and we assume that no incoming traffic class is starved of shaping buffer capacity. Our primary goal is to find an appropriate shaping buffer size B_S so that higher connection carrying capability can be achieved by using statistical shaping than that by using lossless shaping. We search for the appropriate value of B_S through analyzing how the results for single traffic class in section 5.3.1 can be extended to the case of multiple traffic

classes. We shall analyze the admissible region in the heterogeneous environment and show the impact of various shaping schemes on the admissible region.

From the critical values B_{Scrh} and B_{Scrl} of several typical video traces, we find that B_{Scrl} is usually around $10^2 \sim 10^3$ cells and B_{Scrh} is around $10^4 \sim 10^5$ cells. Therefore, it is reasonable to assume that $\min_{1\le j\le J}(B_{Scrh,j}) > \max_{1\le j\le J}(B_{Scrl,j})$.

When lossless shaping is used (i.e., $L_S = 0$), we let $A(L_S = 0, L_M = L)$ and $A(L_S = 0, L_M = 0)$ represent the admissible region when loss probability L is allowed at the multiplexer and when lossless service is guaranteed at the multiplexer, respectively. Then we can prove the following lemma.

Lemma 5.1: When lossless shaping is used (i.e., $L_S = 0$), we let $C_{C0,j} = e_{00,j}\log(\frac{1}{L})/\log(\frac{1}{\chi_j})$. It then follows:

(1) If $C < \min_{1\le j\le J}(C_{C0,j})$, then $A(L_S = 0, L_M = L) = A(L_S = 0, L_M = 0)$, i.e., $A(L_S = 0, L_M = L)$ and $A(L_S = 0, L_M = 0)$ coincide with each other;

(2) If $C > \max_{1\le j\le J}(C_{C0,j})$, then $A(L_S = 0, L_M = L) \supset A(L_S = 0, L_M = 0)$, i.e., $A(L_S = 0, L_M = L)$ subsumes $A(L_S = 0, L_M = 0)$;

(3) If $\min_{1\le j\le J}(C_{C0,j}) \le C \le \max_{1\le j\le J}(C_{C0,j})$, then $A(L_S = 0, L_M = L) \supseteq A(L_S = 0, L_M = 0)$, i.e., their boundaries coincide in part and elsewhere $A(L_S = 0, L_M = L)$ subsumes $A(L_S = 0, L_M = 0)$. The boundaries of $A(L_S = 0, L_M = L)$ are non-linear in this case.

Proof: see Appendix C. 2.

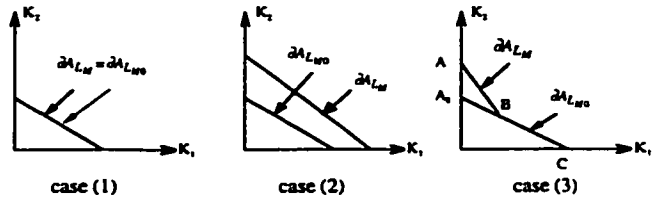


Fig. 5-3 Illustration of Lemma 5.1 When J = 2

Example of Lemma 5.1:

We use an example of two traffic classes (i.e. J = 2) to illustrate the results of Lemma 5.1. As shown in Fig. 5-3, the example consists of three cases. In the figure, ∂A_{L_M} and $\partial A_{L_{M0}}$ denote the boundaries of $A(L_S = 0, L_M = L)$ and $A(L_S = 0, L_M = 0)$, respectively.

Case (1) If $C < \min(C_{C0,1}, C_{C0,2})$, then $A(L_S = 0, L_M = L) = A(L_S = 0, L_M = 0)$;

Case (2) If $C > \max(C_{C0,1}, C_{C0,2})$, then $A(L_S = 0, L_M = L) \supset A(L_S = 0, L_M = 0)$;

Case (3) If $C_{C0,1} \le C \le C_{C0,2}$, then $A(L_S = 0, L_M = L) \supseteq A(L_S = 0, L_M = 0)$, i.e., their boundaries coincide in the line segment BC and elsewhere $A(L_S = 0, L_M = L)$ subsumes $A(L_S = 0, L_M = 0)$. The boundaries of $A(L_S = 0, L_M = L)$ are non-linear in this case, which consisting of two line segments AB and BC. The coordinates of the flex point B ($K_{1,B}$, $K_{2,B}$) can be obtained as follows:

$$\begin{cases} K_{1,B} \cdot e_{00,1} + K_{2,B} \cdot e_{00,2} = C \\ K_{1,B} \cdot \log(1/\chi_1) + K_{2,B} \cdot \log(1/\chi_2) = \log(1/L) \end{cases} . \qquad (5\text{-}7)$$

Similarly, when statistical shaping is used (i.e., $L_S > 0$) , we let $A(L_S > 0, L_M = L - L_S)$ and $A(L_S > 0, L_M = 0)$ denote the admissible region when loss probability $L - L_S$ is allowed at the multiplexer and that when lossless service is guaranteed at the multiplexer, respectively. Then, we have the following lemma.

Lemma 5.2: When statistical shaping is used (i.e., $L_S > 0$), we let $C_{CLs,j} = e_{0L_S,j} \log(\frac{1}{L - L_S}) / \log(\frac{1}{\chi_j})$. It then follows:

(1) If $C < \min_{1 \le j \le J}(C_{CLs,j})$, then $A(L_S > 0, L_M = L - L_S) = A(L_S > 0, L_M = 0)$;

(2) If $C > \max_{1 \le j \le J}(C_{CLs,j})$, then $A(L_S > 0, L_M = L - L_S) \supset A(L_S > 0, L_M = 0)$;

(3) If $\min_{1\le j\le J}(C_{CLs,j}) \le C \le \max_{1\le j\le J}(C_{CLs,j})$, then $A(L_S > 0, L_M = L - L_S) \supseteq A(L_S > 0, L_M = 0)$ and the boundaries of $A(L_S > 0, L_M = L - L_S)$ are non-linear.

Proof: see Appendix C. 3.

From Lemmas 5.1 and 5.2, we know that if $A(L_S = 0, L_M = L)$ and $A(L_S > 0, L_M = L - L_S)$ have non-linear boundaries, part of their boundaries coincide with the boundaries of $A(L_S = 0, L_M = 0)$ and $A(L_S > 0, L_M = 0)$, respectively. Therefore, the previous results about the single class case can not be carried over directly to the case of multiple traffic classes. To make exposition clear, we assume in the following proposition that both $A(L_S = 0, L_M = L)$ and $A(L_S > 0, L_M = L - L_S)$ have linear boundaries. The case of non-linear boundaries will be illustrated using an example at the end of this section.

Proposition 5.2: Assume that both $A(L_S = 0, L_M = L)$ and $A(L_S > 0, L_M = L - L_S)$ have linear boundaries, and that both values of B_M and C are fixed.

(1) If $B_S \le \min_{1\le j\le J}(B_{Scrl,j})$, $A(L_S>0, L_M=L-L_S) \subseteq A(L_S=0, L_M=L)$, i.e., $A(L_S>0, L_M=L-L_S)$ and $A(L_S=0, L_M=L)$ may coincide at certain corner points and elsewhere $A(L_S = 0, L_M = L)$ subsumes $A(L_S>0, L_M=L-L_S)$;

(2) If $\min_{1\le j\le J}(B_{Scrl,j}) < B_S \le \max_{1\le j\le J}(B_{Scrl,j})$, $A(L_S>0, L_M=L-L_S)$ X $A(L_S=0, L_M=L)$, i.e., the boundaries of $A(L_S>0, L_M=L-L_S)$ and $A(L_S=0, L_M=L)$ cross with each other;

(3) If $\max_{1\le j\le J}(B_{Scrl,j}) < B_S < \min_{1\le j\le J}(B_{Scrh,j})$, $A(L_S>0, L_M=L-L_S) \supset A(L_S=0, L_M=L)$, i.e., $A(L_S>0, L_M=L-L_S)$ subsumes $A(L_S=0, L_M=L)$;

(4) If $\min_{1\le j\le J}(B_{Scrh,j}) \le B_S < \max_{1\le j\le J}(B_{Scrh,j})$, $A(L_S>0, L_M=L-L_S) \supseteq A(L_S=0, L_M=L)$, i.e., $A(L_S>0, L_M=L-L_S)$ and $A(L_S=0, L_M=L)$ may coincide at certain corner points and elsewhere $A(L_S>0, L_M=L-L_S)$ subsumes $A(L_S=0, L_M=L)$;

(5) If $B_S \ge \max_{1\le j\le J}(B_{Scrh,j})$, $A(L_S>0, L_M=L-L_S) = A(L_S=0, L_M=L)$, i.e., $A(L_S>0, L_M=L-L_S)$ and $A(L_S=0, L_M=L)$ coincide with each other.

Proof: see Appendix C. 4.

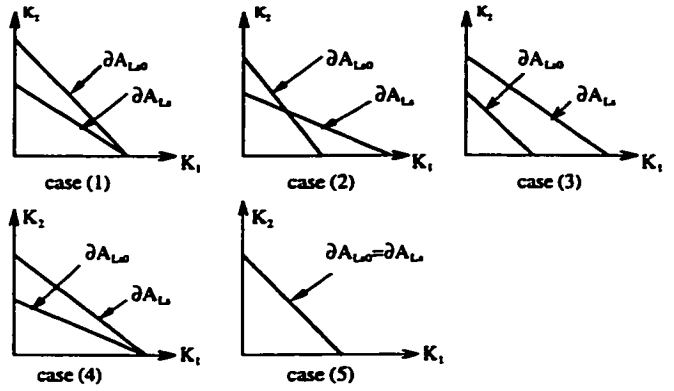


Fig. 5-4 Illustration of Proposition 5.2 When J = 2

Example of Proposition 5.2:

When J = 2, the above proposition can be illustrated using Fig. 5-4, which consists of five cases. In the figure, ∂A_{Ls} and ∂A_{Ls0} denote the boundaries of $A(L_S>0, L_M=L-L_S)$ and $A(L_S=0, L_M=L)$, respectively.

Case (1) If $B_S \leq \min(B_{Scrl,1}, B_{Scrl,2})$, $A(L_S > 0, L_M = L - L_S) \subseteq A(L_S = 0, L_M = L)$;

Case (2) If $B_{Scrl,1} < B_S \leq B_{Scrl,2}$, $A(L_S > 0, L_M = L - L_S)$ X $A(L_S = 0, L_M = L)$;

Case (3) If $\max(B_{Scrl,1}, B_{scrl,2}) < B_S < \min(B_{Scrh,1}, B_{Scrh,2})$, $A(L_S > 0, L_M = L - L_S) \supset A(L_S = 0, L_M = L)$;

Case(4) If $B_{Scrh,1} \leq B_S < B_{Scrh,2}$, $A(L_S > 0, L_M = L - L_S) \supseteq A(L_S = 0, L_M = L)$;

Case (5) If $B_S \geq \max(B_{Scrh,1}, B_{Scrh,2})$, $A(L_S > 0, L_M = L - L_S) = A(L_S = 0, L_M = L)$.

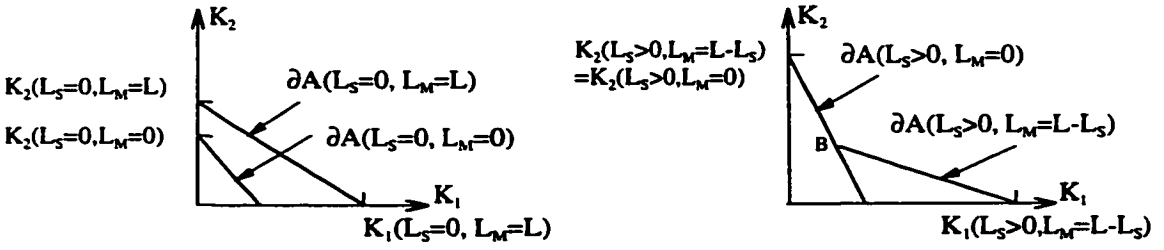
From Fig. 5-4, it can be seen that statistical shaping is not more beneficial for increasing the connection carrying capability than lossless shaping in cases (1), (2), and (5). In case (4), $A(L_S > 0, L_M = L - L_S)$ and $A(L_S = 0, L_M = L)$ coincide at a corner point and elsewhere $A(L_S > 0, L_M = L - L_S)$ subsumes $A(L_S = 0, L_M = L)$, thus statistical shaping is better than lossless shaping only in part. In case (3), $A(L_S=0, L_M=L)$ is a proper subset of $A(L_S > 0, L_M = L - L_S)$, so statistical shaping is absolutely more beneficial than lossless shaping in this case. Clearly, when B_S is

chosen within a right interval, a larger admissible region can be obtained by using statistical shaping, i.e., a higher level of resource utilization gain can be achieved.

In Proposition 5.2, we assume that both $A(L_S = 0, L_M = L)$ and $A(L_S > 0, L_M = L - L_S)$ have linear boundaries. When either one or both of them have non-linear boundaries, there are much more possible cases than that stated in Proposition 5.2. The results are not listed here due to space limitation; they can be extended easily from the above Lemma 5.1, Lemma 5.2, and Proposition 5.2. The following example is used to illustrate the case of non-linear boundaries.

Example of non-linear boundaries:

When J = 2, we assume that $C > \max(C_{C0,1}, C_{C0,2})$ and $C_{CLs,1} < C < C_{CLs,2}$, then $A(L_S = 0, L_M = L)$ has linear boundaries while $A(L_S > 0, L_M = L - L_S)$ has non-linear boundaries, as shown in Fig. 5-5. If $\max(B_{Scrl,1}, B_{Scrl,2}) < B_S < \min(B_{Scrh,1}, B_{Scrh,2})$, we know from Proposition 5.1 that $K_1(L_S > 0, L_M = L - L_S) > K_1(L_S = 0, L_M = L)$ and that $K_2(L_S > 0, L_M = L-L_S) > K_2(L_S = 0, L_M = L)$. But $A(L_S > 0, L_M = L - L_S)$ has non-linear and convex boundaries. Thus, the coordinates of the flex point B need to be obtained by using eqn. (5-7) in order to compare $A(L_S = 0, L_M = L)$ and $A(L_S > 0, L_M = L - L_S)$. We can see that the previous results in the contents of single traffic class can not be carried over directly to the heterogeneous environment when the boundaries of the admissible region are non-linear.

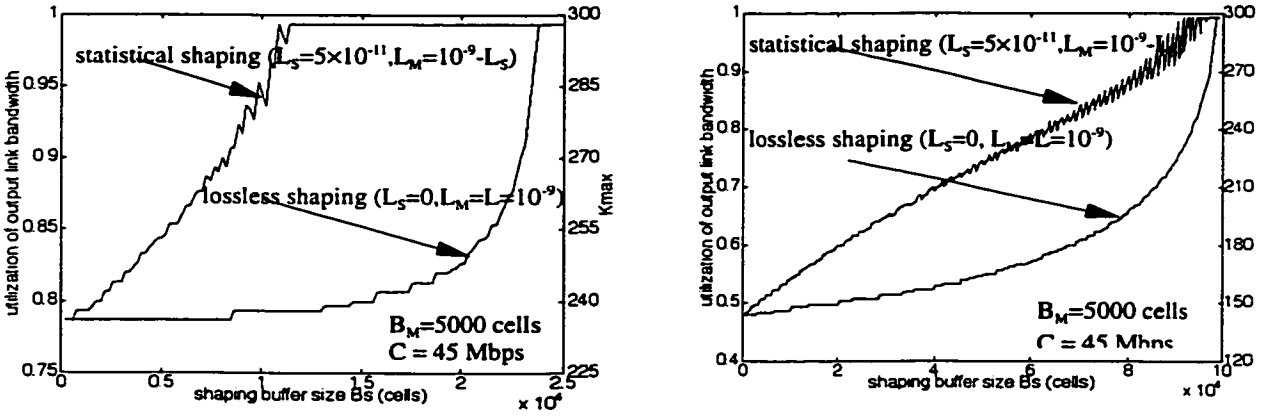


(i) when $L_S = 0$ (ii) when $L_S > 0$

Fig. 5-5 An Example of the Non-Linear Boundary Case

5.4 Performance Evaluation

Performance evaluation results are presented in this section to illustrate the analysis results of earlier sections. Two typical MPEG video traffic flows different from Chapter 3 are used here; both are characterized using three standard traffic parameters. The parameters of type-1 incoming traffic are: peak rate $\tilde{P}_1$ = 6 Mb/s, burst size $\tilde{\sigma}_1$ = 10.6 Kbits, and long-term-average-rate $\tilde{\rho}_1$= 150 Kb/s. The parameters of type-2 traffic are: $\tilde{P}_2$ = 2 Mb/s, $\tilde{\sigma}_2$ = 42.4 Kbits, and $\tilde{\rho}_2$ = 150 Kb/s. In the following experiments, the number of traffic flows traversing the shaper is fixed as K_{S0}= 1000.



type-1 traffic type-2 traffic

Fig. 5-6 Bandwidth Utilization vs. the Shaping Buffer Size

In the first experiment, we study the case of homogeneous traffic class, i.e., either type-1 or type-2 traffic flows are input into our network node model. We investigate the impact of shaping buffer size on the average utilization of the output link bandwidth. The average utilization is calculated as $K_{max} \cdot \tilde{\rho}$/(link capacity), where, K_{max} is the maximum number of

admissible connections, and $\tilde{\rho}$ is the long-term-average-rate of the incoming traffic. The results are shown in Fig. 5-6. For the case that all incoming traffic flows are of type-1, when 410 cells < B_S < 23809 cells, statistical shaping achieves higher bandwidth utilization than lossless shaping; otherwise, statistical shaping is no longer beneficial for improving link bandwidth utilization. Similarly, for the case of type-2 traffic flows alone, when 670 cells <B_S <98521 cells, statistical shaping is more beneficial for improving the utilization of link bandwidth than lossless shaping. Thus, we can see that the critical values of B_S discussed in section 5.3.1 do exist. By using eqn. (5-6) and Procedure *find_B_{Scrl}* in Appendix C.1, we obtain that the critical values of the shaping buffer capacity for type-1 traffic are $B_{Scrh,1}$ = 23809 cells and $B_{Scrl,1}$ = 410 cells, and that the critical values for type-2 traffic are $B_{Scrh,2}$ = 98521 cells and $B_{Scrl,2}$ = 670 cells. These computed values are just consistent with the results of experiment shown in Fig. 5-6.

From Fig. 5-6, one can see that after B_S becomes larger than certain value, the bandwidth utilization no longer increases with B_S. This is because, as analyzed in section 5.3.1, when B_S is so large that $e_{00} = \tilde{\rho}$, the bandwidth utilization can not be further improved by increasing B_S. We also see that the predominant trend is that the bandwidth utilization increases with the increasing value of B_S, but there is oscillation. The oscillation is due to that a small increase in B_S does not always increase the number of admissible connections, as analyzed in section 5.3.1.

Comparing the above two diagrams, one can see that for a given B_S value, the achievable utilization is lower for the type-2 traffic than that for the type-1 traffic. This is because the type-2 traffic is more bursty (e.g., has larger burst size and converges slower to the average rate) than the type-1 traffic.

If comparing Fig. 5-6 with Fig. 3-8 in Chapter 3, we see that there is a big difference between the bandwidth utilization depicted in these two figures. Please note that these two

diagrams are obtained under very different situations. In Fig. 3-8, no cell loss is allowed and no traffic shaping is used. However in Fig. 5-6, traffic shaping is used and the cell loss is allowed at both the shaper and multiplexer. When a big shaping buffer is available as in Fig. 5-6, the incoming traffic flows can be smoothed out with its peak rate close to its sustainable rate. In this way, a higher level of bandwidth utilization is achieved. However, this benefit comes at the price of a larger jitter and a requirement for a larger backup buffer at the receiver. In addition, we can see that bandwidth utilization varies with the shaping buffer size, therefore it is hard to say precisely how much percentage utilization can be improved by statistical shaping.

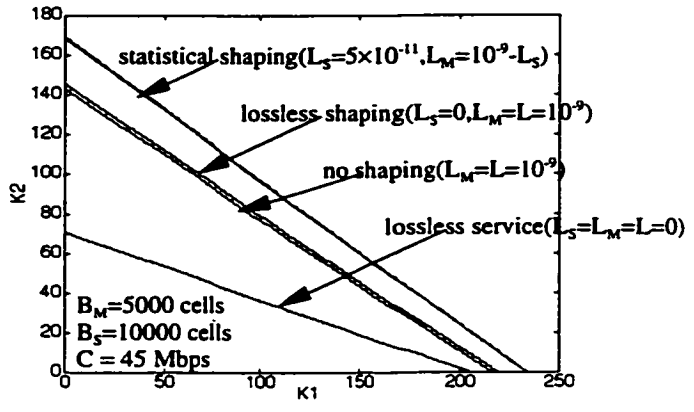


Fig. 5-7 The Admissible Region

Next, we study the case of multiple traffic classes. The traffic flows of both type-1 and type-2 are input into our network node model of Fig. 2-2. The shaping buffer is completely shared among these two types of traffic flows. We evaluate the bandwidth utilization in the heterogeneous environment by using the admissible region, since the admissible region boundaries in Fig. 5-7 essentially give the maximum number of admissible connections of both type-1 and type-2 traffic flows. There are several noteworthy points about Fig. 5-7:

(1) Based on the definitions in Lemma 5.1, we obtained $C_{C0,1}$=12.1 Mb/s and $C_{C0,2}$=9.1 Mb/s. Based on Lemma 5.2, we obtained for $L_S = 5 \times 10^{-11}$ that $C_{CLs,1}$=16.3 Mb/s and $C_{CLs,2}$=8.6 Mb/s. We choose the output link capacity as C = 45 Mbps, thus $C > \max(C_{C0,1}, C_{C0,2})$ and $C > \max(C_{CLs,1}, C_{CLs,2})$. Just being consistent with Lemmas 5.1 and 5.2, those admissible regions shown in Fig. 5-7 have linear boundaries.

(2) When small loss probability is allowed at the network node, the connection carrying capability can be increased considerably than that when lossless service is guaranteed.

(3) When cell loss is allowed *only* at the multiplexer, more connections can be admitted when lossless shaping is used than that when no traffic shaping is used, since larger statistical multiplexing gain can be achieved when multiplexing less bursty traffic after traffic shaping.

(4) When statistical shaping is used, the connection carrying capability can be further improved than that when lossless shaping is used. This is consistent with Proposition 5.2, i.e., when $\max(B_{Scrl,1}, B_{Scrl,2}) < B_S < \min(B_{Scrh,1}, B_{Scrh,2})$, we have $A(L_S>0, L_M=L-L_S) \supset A(L_S=0, L_M=L)$. Therefore, when the shaping buffer capacity B_S is within a right interval, allowing cell loss at both the shaper and multiplexer is more beneficial for improving the network resource utilization than allowing cell loss only at the multiplexer.

5.5 Chapter Summary and Remarks

We have demonstrated in this chapter how to improve the resource utilization in the context of a network node applying the rate-controlled service (RCS) discipline. We proposed to allow a small loss probability at the traffic shaper, and analyzed the impact of traffic shaping schemes on the network resource utilization. Our analysis established that there exists condition under which our proposed statistical shaping scheme is more beneficial than lossless shaping for

improving resource utilization. Some design criteria were also obtained from the analysis. Performance evaluation showed that statistical traffic shaping can achieve a higher level of resource utilization than lossless traffic shaping when we apply these criteria in the network design.

Chapter 6

Impact of Rate-controlling on End-to-end QoS Provisioning

6.1 Introduction

In this chapter, we investigate the impact of rate-controlling on end-to-end QoS provisioning through analyzing the statistical end-to-end performance of an individual connection. Analysis is carried out for the network model of section 2.1. For the ease of reference, we repeat the network model in Fig. 6-1, and we also indicate in it the traffic parameters of both the reference traffic and the background traffic.

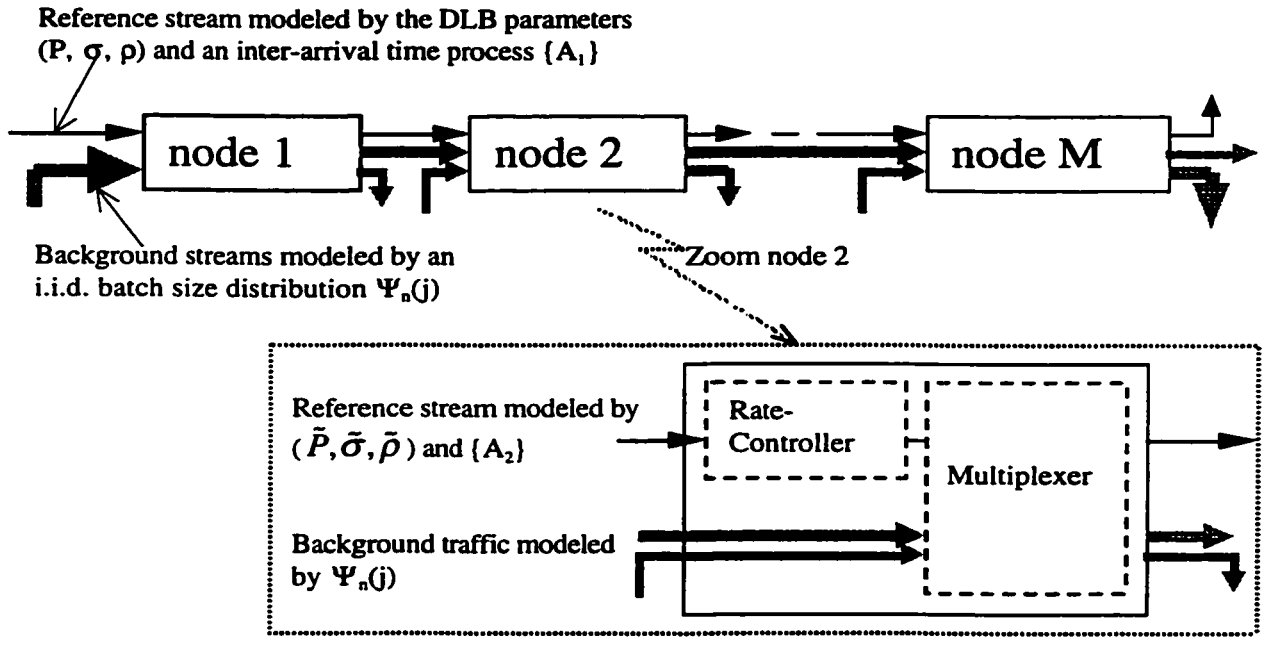


Fig. 6-1 Network Model

At a network node, the reference stream is shaped at a leaky-bucket rate-controller before entering the multiplexer. The reference stream at the network entrance is specified by a set of DLB parameters: the peak rate P, average rate ρ, and the maximum burst size σ. Besides, its cell inter-arrival time follows the generally distributed process $\{A_k\}_{k\geq 1}$ of section 2.2.3. However, as stated in section 2.2.2, the reference stream's traffic pattern could be distorted by the interaction with background streams when they are switched at a multiplexer. Thus at a node inside the network, the parameter set $(\tilde{P},\tilde{\sigma},\tilde{\rho})$ is used instead to characterize the incoming traffic at a rate-controller. The background traffic is modeled by using the i. i. d. batch arrival process of section 2.2.4.

There are three tasks involved in our analysis for the end-to-end QoS of the reference stream. First, we shall analyze for the queuing performance at the rate-controller (section 6.2). Second, the performance at the multiplexer will be studied (section 6.3). Third, based on the analysis results for both the rate-controller and multiplexer, the end-to-end performance will be obtained (section 6.4). Furthermore, performance evaluation will be conducted to investigate the impact of rate-controlling on the end-to-end QoS (section 6.5).

6.2 Queuing Analysis of Rate-controller

Under the assumption of a generally distributed inter-arrival time process, we analyze the statistical performance of the rate-controller, including the cell delay distribution and the inter-departure time process. First, a new operating mechanism is proposed for a leaky bucket rate-controller. The new mechanism can fulfill the same function as the "classical" leaky bucket scheme, but it has the major advantage of allowing a simple embedded Markov chain analysis

for the rate-controller. Then, analysis is performed to study the statistical performance of the rate-controller.

6.2.1 Operating Mechanism

The "classical" Leaky Bucket scheme has been studied in many previous works. The operating mechanism of the "classical" Leaky Bucket scheme can be found in Appendix D.1.

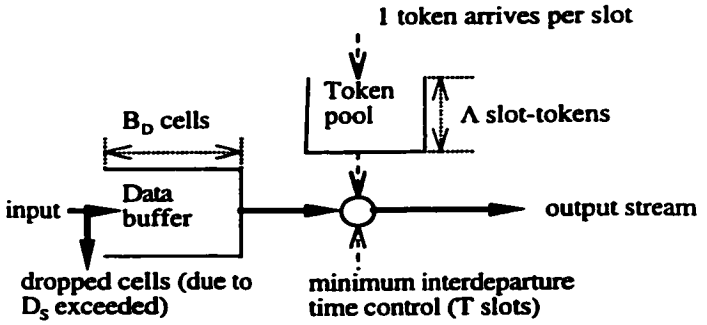


Fig. 6-2 Illustration of the Proposed Leaky Bucket Mechanism

We propose here a new leaky bucket mechanism. With the aid of Fig. 6-2, the rules for the proposed mechanism are described as follows:

(a) A slot-token is used instead of the "classical" token. The slot-token is so called since it is generated at the rate of one token per time-slot. A slot-token pool of capacity Λ. is available. The slot-token is generated just prior to the end of a slot. The newly generated slot-token is stored in the pool if it is not full; otherwise the newly generated slot-token is lost.

(b) The data buffer maintains a FIFO queue. Packet arrivals are considered at the end of a slot. If the queue is not empty, the packet at the head of the queue can be transmitted only when there are at least Γ slot-tokens in the token pool and at least T time-slots have passed since the last departure. One cell transmission must be accompanied with the removal of Γ slot-

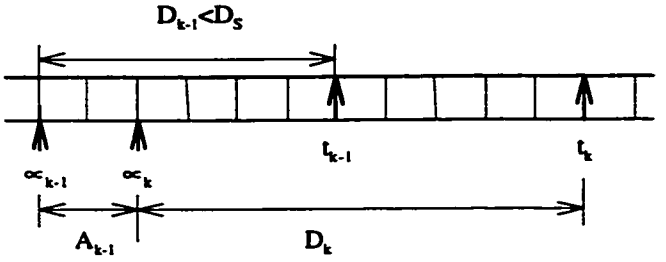
tokens from the pool. Based on the above rules, the waiting time of an arriving cell can be obtained right after its arrival (as detailed in section 6.2.2). Because the cell delay is an important QoS measure for real-time applications, we propose to drop an arriving cell when its waiting time exceeds a certain delay bound D_S rather than when the data buffer is full. That is, if the waiting time of an arriving packet is longer than a certain delay bound D_S, the packet will be dropped immediately at the entrance; otherwise it will enter the data buffer and wait there until its waiting time has passed.

The proposed leaky bucket has several interesting and important aspects. First, we can make the data buffer big enough (which is not a serious limitation nowadays), so that the only reason that a cell is discarded is because its delay bound is exceeded. For example, we can select $B_D = \max\{ \lceil D_S / T \rceil, \lceil (D_S+\Lambda) / \Gamma \rceil \}$, where $\lceil x \rceil$ is the smallest integer larger than or equal to x. Under the operation above and assuming that the slot-token can be consumed by any new arrivals, then it is easy to verify that the proposed mechanism with parameters $(T, \Gamma, \Lambda, B_D, D_S)$ is equivalent to the "classical" one with parameters $(P_S, \beta, \gamma, B_D)$ when $T = 1 / P_S$, $\Gamma = 1 / \gamma$, $\Lambda = \beta/\gamma$, and $D_S \geq \max\{ \lceil (B_D - \beta) / \gamma \rceil, \lceil B_D / P_S \rceil \}$.

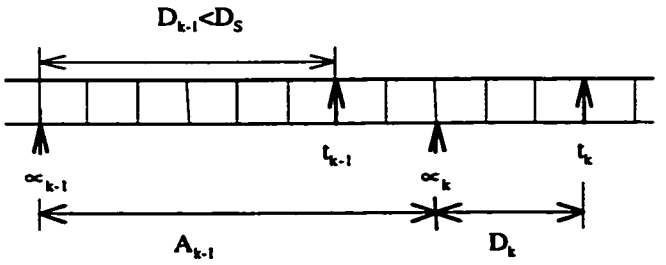
Our model has a further feature of handling packet discarding due to the exceeded delay bound, which is of common interest to the QoS provisioning in high speed networks. This is a case that the "classical" mechanism does not handle because the delay bound is not usually "defined" in the "classical" mechanism. As shall be demonstrated in the following section, our model facilitates a simple analysis. The case that our model may not handle is the overflow of the data buffer, which is not a serious limitation as we mentioned before.

6.2.2. Cell delay distribution

We derive here the cell delay distribution at the rate-controller. Let $\propto_k$ denote the arrival time of the kth cell, which is the end of the slot over which the kth cell is completely received by the leaky bucket. Let t_k denote the departure time of the kth cell. Then $D_k = t_k - \propto_k$ is the delay of the kth cell at the rate-controller. Clearly, the process $\{D_k\}_{k\geq 1}$ does not form a Markov chain, because the knowledge of the value of D_k alone is not sufficient to determine the probability distribution of D_{k+1}. In order to cope with this difficulty, we let the random variable S_k indicate the token pool occupancy immediately after the kth cell's service epoch t_k. It is easy to see that the queuing system can be described by a two-dimensional Markov chain $\{D_k, S_k\}_{k\geq 1}$ embedded at the service epoch $\{t_k\}_{k\geq 1}$.

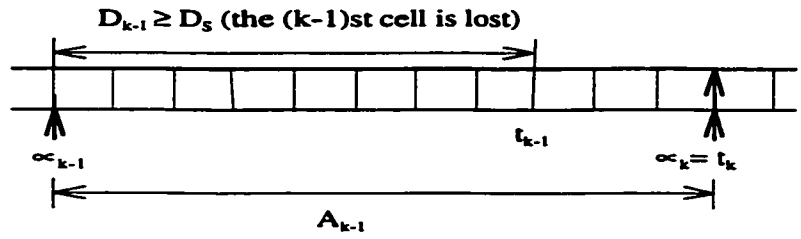


(a) When ($D_{k-1} < D_S$) and ($A_{k-1} \leq D_{k-1}$)

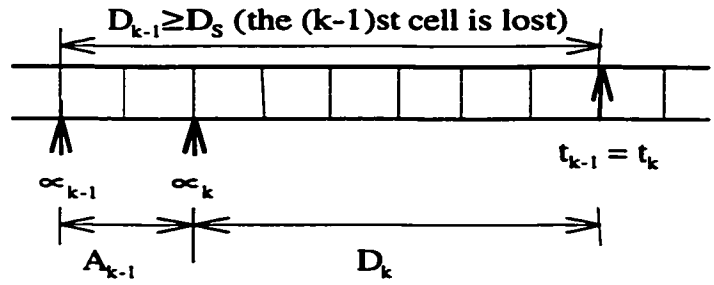


(b) When ($D_{k-1} < D_S$) and ($A_{k-1} > D_{k-1}$)

Fig. 6-3 Delay Analysis When $D_{k-1} < D_S$



(a) When ($D_{k-1} \geq D_S$) and ($A_{k-1} > D_{k-1}$)



(b) When ($D_{k-1} \geq D_S$) and ($A_{k-1} \leq D_{k-1}$)

Fig. 6-4 Delay Analysis When $D_{k-1} \geq D_S$

With reference to Fig. 6-3 when $D_{k-1} < D_S$, and with reference to Fig. 6-4 when $D_{k-1} \geq D_S$, we derived the expressions for D_k and S_k as follows:

$$D_k = \begin{cases} 0 & \text{if } (0 \leq D_{k-1} < D_S) \text{ and } (0 \leq S_{k-1} \leq \Lambda) \text{ and } (A_{k-1} \geq D_{k-1} + \max\{T, \Gamma - S_{k-1}\}) \\ D_{k-1} + T - A_{k-1} & \text{if } (0 \leq D_{k-1} < D_S) \text{ and } (\Gamma - T \leq S_{k-1} \leq \Lambda) \text{ and } (1 \leq A_{k-1} < D_{k-1} + T) \\ D_{k-1} + \Gamma - S_{k-1} - A_{k-1} & \text{if } (0 \leq D_{k-1} < D_S) \text{ and } (0 \leq S_{k-1} < \Gamma - T) \text{ and } (1 \leq A_{k-1} < D_{k-1} + \Gamma - S_{k-1}) \\ 0 & \text{if } (D_{k-1} \geq D_S) \text{ and } (A_{k-1} > D_{k-1}) \\ D_{k-1} - A_{k-1} & \text{if } (D_{k-1} \geq D_S) \text{ and } (1 \leq A_{k-1} \leq D_{k-1}) \end{cases} \quad (6\text{-}1)$$

$$S_k = \begin{cases} \min\{S_{k-1} + A_{k-1} - D_{k-1}, \Lambda\} - \Gamma & \text{if } (0 \leq D_{k-1} < D_S) \text{ and } (0 \leq S_{k-1} \leq \Lambda) \text{ and } (A_{k-1} \geq D_{k-1} + \max\{T, \Gamma - S_{k-1}\}) \\ S_{k-1} + T - \Gamma & \text{if } (0 \leq D_{k-1} < D_S) \text{ and } (\Gamma - T \leq S_{k-1} \leq \Lambda) \text{ and } (1 \leq A_{k-1} < D_{k-1} + T) \\ 0 & \text{if } (0 \leq D_{k-1} < D_S) \text{ and } (0 \leq S_{k-1} < \Gamma - T) \text{ and } (1 \leq A_{k-1} < D_{k-1} + \Gamma - S_{k-1}) \\ \min\{S_{k-1} + \Gamma + A_{k-1} - D_{k-1}, \Lambda\} - \Gamma & \text{if } (D_{k-1} \geq D_S) \text{ and } (A_{k-1} > D_{k-1}) \\ S_{k-1} & \text{if } (D_{k-1} \geq D_S) \text{ and } (1 \leq A_{k-1} \leq D_{k-1}) \end{cases} \quad (6\text{-}2)$$

The above equations (6-1) and (6-2) indicate that, for k >1, both D_k and S_k can be computed at time $\propto_k$, using the values A_{k-1}, D_{k-1}, and S_{k-1}. For k = 1, with a full token pool and an empty data buffer at the beginning of operation, we have $D_1 = 0$ and $S_1 = \Lambda - \Gamma$. It can also be seen that there are five cases in each of equations (6-1) and (6-2). In cases (1) - (3), $0 \le D_{k-1} \le D_S$, which means that the *(k-1)*st cell is not discarded. In both cases (4) and (5), $D_{k-1} > D_S$, which means that *(k-1)*st cell is discarded due to the exceeded delay bound.

In case (1), we have ($0 \le S_{k-1} \le \Lambda$), ($A_{k-1} \ge D_{k-1} + \max\{T, \Gamma - S_{k-1}\}$) and ($0 \le D_{k-1} < D_S$). The inequality $A_{k-1} \ge D_{k-1} + \max\{T, \Gamma - S_{k-1}\}$ means that, at the arriving time of the *k*th cell, there are at least Γ slot-tokens in the pool and at least T time-slots have passed since the departure of the *(k-1)*st cell. Thus, the *k*th cell can be transmitted immediately after its arrival, i.e., $D_k = 0$. It is easy to show that $S_k = \min\{S_{k-1} + A_{k-1} - D_{k-1}, \Lambda\} - \Gamma$ for this case.

In case (2), we have ($\Gamma - T \le S_{k-1} \le \Lambda$), ($1 \le A_{k-1} < D_{k-1} + T$) and ($0 \le D_{k-1} < D_S$). The inequality $A_{k-1} < D_{k-1} + T$ means that, at the *k*th cell arriving time, less than T slots have passed since the departure of the *(k-1)*st cell. The inequality $S_{k-1} \ge \Gamma - T$ means that there are at least Γ slot-tokens in the pool at the time when T slots have passed since the departure of the *(k-1)*st cell. Thus, the *k*th cell will be served at the time $t_{k-1} + T$. Therefore, we have $S_k = S_{k-1} + T - \Gamma$ and $D_k = D_{k-1} + T - A_{k-1}$.

In case (3), we have ($S_{k-1} < \Gamma - T$), ($1 \le A_{k-1} < D_{k-1} + \Gamma - S_{k-1}$) and ($D_{k-1} < D_S$). The inequality $S_{k-1} < \Gamma - T$ means that there are less than Γ slot-tokens in the pool at the time $t_{k-1} + T$. Two inequalities ($A_{k-1} < D_{k-1} + \Gamma - S_{k-1}$) and ($S_{k-1} < \Gamma - T$) together indicate that there are less than Γ slot-tokens in the pool when the *k*th cell arrives. Therefore, the *k*th cell will wait in the buffer,

and it will be served as soon as there are Γ slot-tokens in the pool. So, we have $S_k = 0$ and $D_k = D_{k-1} + \Gamma - S_{k-1} - A_{k-1}$.

In case (4), we have $(A_{k-1} > D_{k-1})$ and $(D_{k-1} \geq D_S)$. The inequality $D_{k-1} \geq D_S$ means that the *(k-1)*st cell is dropped at the entrance of the buffer due to the exceeded delay bound. This inequality also means that it is until t_{k-1} that there are less than Γ slot-tokens in the pool or that less than T slots have passed since the last departure. The inequality $A_{k-1} > D_{k-1}$ means that the *k*th cell arrives after t_{k-1}; therefore, it can be transmitted immediately after its arrival. So, we have $D_k = 0$ and $S_k = \min\{S_{k-1} + \Gamma + A_{k-1} - D_{k-1}, \Lambda\} - \Gamma$.

In case (5), we have $(A_{k-1} \leq D_{k-1})$ and $(D_{k-1} \geq D_S)$. The inequality $A_{k-1} \leq D_{k-1}$ means that the *k*th cell arrives before t_{k-1}. Based on the explanation of $D_{k-1} \geq D_S$ in case (4), we know that the *k*th cell will not be served until t_{k-1}. Thus, we have $D_k = D_{k-1} - A_{k-1}$ and $S_k = S_{k-1}$.

Let $(D_{k-1}, S_{k-1}) = (j, i)$ and $(D_k, S_k) = (j', i')$. Based on the discussion of the five cases in equations (6-1) and (6-2), the transition probabilities for $\{D_k, S_k\}_{k\geq 1}$, denoted by p(j, i; j', i') can be analyzed into the following six cases.

(1) When $(0 \leq j < D_S)$ and $(0 \leq i \leq \Lambda)$ and $(j' = 0)$ and $(\min\{i + \max(T, \Gamma - i), \Lambda\} - \Gamma \leq i' \leq \Lambda - \Gamma)$

$$p(j, i; j', i') = \begin{cases} a(i' - i + \Gamma + j) & \text{if } i' < \Lambda - \Gamma \\ \sum_{n=\Lambda+j-i}^{A\max} a(n) & \text{if } i' = \Lambda - \Gamma \end{cases} .$$

(2) When $(0 \leq j < D_S)$ and $(\Gamma - T \leq i \leq \Lambda)$ and $(0 < j' < j + T)$ and $(i' = i + T - \Gamma)$

$p(j, i; j', i') = a(j + T - j')$.

(3) When $(0 \leq j < D_S)$ and $(0 \leq i \leq \Gamma - T)$ and $(0 < j' < j + \Gamma - i)$ and $(i' = 0)$

$p(j, i; j', i') = a(j + \Gamma - j' - i)$.

(4) When $(j \geq D_S)$ and $(j' = 0)$ and $(\min\{i + \Gamma, \Lambda\} - \Gamma \leq i' \leq \Lambda - \Gamma)$

$$p(j, i; j', i') = \begin{cases} a(j + i' - i) & \text{for } i' < \Lambda - \Gamma \\ \sum_{n=\Lambda+j-i-\Gamma}^{A_{max}} a(n) & \text{for } i' = \Lambda - \Gamma \end{cases} .$$

(5) When $(j \geq D_S)$ and $(0 \leq j' < j)$ and $(i' = i)$

$p(j, i; j', i') = a(j - j')$.

(6) Otherwise, $p(j, i; j', i') = 0$.

Thus, the transitional probability matrix $\boldsymbol{P}$ of the Markov chain $\{D_k, S_k\}_{k \geq 1}$ can be obtained by assigning the value of $p(j, i; j', i')$ to a corresponding matrix element $p(l, l')$, where l and l' are row and column indices of the matrix element $p(l, l')$. It is easy to verify that the maximum value of S_k is $\Lambda - \Gamma$ when $D_k = 0$, and that the maximum value of S_k is $\Lambda + T - 2\Gamma$ when $D_k > 0$. Therefore, we have

$$l = \begin{cases} i + 1 & \text{for } j = 0 \\ (\Lambda - \Gamma + 1) + (j - 1) \cdot (\Lambda - 2\Gamma + T + 1) + (i + 1) & \text{for } 1 \leq j \leq D_{max} \end{cases} ;$$

$$l' = \begin{cases} i' + 1 & \text{for } j' = 0 \\ (\Lambda - \Gamma + 1) + (j' - 1) \cdot (\Lambda - 2\Gamma + T + 1) + (i' + 1) & \text{for } 1 \leq j' \leq D_{max} \end{cases} ;$$

$D_{max} = \Gamma + D_S - 1 - A_{min}$.

It is easy to show that the Markov chain $\{D_k, S_k\}_{k \geq 1}$ is ergodic, so the equilibrium probability $\pi(j,i) = \lim_{k \to \infty} \Pr\{D_k = j, S_k = i\}$ exists if and only if $\Gamma \leq 1 / \tilde{\rho}$, where $\tilde{\rho}$ is the long-term average rate of the input stream. The equilibrium probability $\pi(j,i)$ can be derived by solving

$$\begin{cases} \bar{\pi} = \bar{\pi} \cdot \boldsymbol{P} \\ \bar{\pi} \cdot \bar{e}^T = 1 \end{cases} ,$$

where $\bar{e} = [1, 1, \ldots\ldots, 1]_{1\times N}$;

$\bar{e}^T$ is the transpose of $\bar{e}$;

$N = (\Lambda - \Gamma + 1) + (\Lambda - 2\Gamma + T + 1) \cdot D_{max}$.

Thus, the stationary probability distribution for cell delay at the LB shaper can be obtained as

$$\Pr\{\text{cell delay} = j\} = \begin{cases} \sum_{i=0}^{\Lambda-\Gamma} \pi(j,i) & \text{for } j = 0 \\ \sum_{i=0}^{\Lambda-2\Gamma+T} \pi(j,i) & \text{for } j > 0 \end{cases}$$

6.2.3 The inter-departure process

We derive here the distribution of the inter-departure time between two successive departure cells from the rate-controller. Since the Markov chain $\{D_k, S_k\}_{k\geq 1}$ does not provide enough information to compute the distribution of inter-departure time, we need to describe the system using the Markov chain $\{D_k, S_k, A_k\}_{k\geq 1}$, where $A_k = \propto_{k+1} - \propto_k$ is the inter-arrival time between the $(k+1)$st and the kth arriving cell. Let $\pi_1(j,i,h)$ denote the equilibrium probabilities for this Markov chain. From equations (6-1) and (6-2), we can see that both D_k and S_k are independent of A_k. Thus, the equilibrium probability for the Markov chain $\{D_k, S_k, A_k\}_{k\geq 1}$ can be computed as $\pi_1(j,i,h) = \pi(j,i) \cdot a(h)$, where $a(h) = \Pr\{A_k = h\}$ is the probability distribution of the inter-arrival time process $\{A_k\}_{k\geq 1}$.

Please note that the sequence numbering of the arriving cells may be different from that of the departures, because cell loss may be incurred at the rate-controller. Let X_{n+1} denote the inter-departure time between the nth departure and the $(n+1)$st departure. That is, at the exit of the rate-controller, the nth departing cell is output immediately prior to the $(n+1)$st departing

cell. However, at the entrance of the rate-controller, some other cells may have arrived between these two cells, but they have been dropped due to the exceeded delay bound. Without loss of generality, we can assume that the nth departure is the kth cell in the arriving cell stream, where n and k are arbitrary positive integers with $n \leq k$. Then, the $(n+1)$st departure could be the $(k+1)$st arrival if the $(k+1)$st arriving cell passes the rate-controller successfully (i.e., $D_{k+1} < D_S$). If the $(k+1)$st arrival is dropped, but the $(k+2)$nd arriving cell passes the rate-controller successfully, then the $(k+2)$nd arrival becomes the $(n+1)$st departure. It is easy to verify that at most $(\Gamma-1)$ cells can be dropped consecutively under the proposed mechanism with parameters $(T, \Lambda, \Gamma, B_D)$. Therefore, if the nth departure is the kth cell in the arriving stream, the $(n+1)$st departure could be the $(k+v)$th cell in the arriving stream, where $v = 1, 2, \ldots, \Gamma$. Based on the above analysis, we derive the distribution of the inter-departure time as follows:

$$\Pr\{X_{n+1} = m\}$$

$$= \sum_{u=0}^{\Gamma-1} [\Pr\{X_{n+1} = m \mid u \text{ arrivals have been dropped between the } n\text{th and the } (n+1)\text{st departures}\}$$
$$\cdot \Pr\{u \text{ arrivals have been dropped between the } n\text{th and the } (n+1)\text{st departure}\}]$$

$$= \sum_{u=0}^{\Gamma-1} \Pr\{X_{n+1} = m,\ u \text{ arrivals have been dropped between the } n\text{th and the } (n+1)\text{st departures}\}$$

$$= \Pr\{X_{n+1} = m,\ D_k < D_S,\ D_{k+1} < D_S\}$$
$$+ \Pr\{X_{n+1} = m,\ D_k < D_S,\ D_{k+1} \geq D_S, D_{k+2} < D_S\}$$
$$+ \Pr\{X_{n+1} = m,\ D_k < D_S,\ D_{k+1} \geq D_S, D_{k+2} \geq D_S, D_{k+3} < D_S\}$$
$$+ \ldots\ldots$$
$$+ \Pr\{X_{n+1} = m,\ D_k < D_S,\ D_{k+1} \geq D_S, D_{k+2} \geq D_S, \ldots, D_{k+\Gamma-1} \geq D_S, D_{k+\Gamma} < D_S\}$$

$$= \sum_{j=0}^{D_S-1} \sum_{j'=0}^{D_S-1} \Pr\{X_{n+1} = m, D_k = j, D_{k+1} = j'\}$$

$$+ \sum_{j=0}^{D_S-1} \sum_{j'=D_S}^{D_{\max}} \sum_{l_1=0}^{D_S-1} \Pr\{X_{n+1} = m, D_k = j, D_{k+1} = j', D_{k+2} = l_1\}$$

$$+\sum_{j=0}^{D_S-1}\sum_{j'=D_S}^{D_{max}}\sum_{l_1=D_S}^{D_{max}}\sum_{l_2=0}^{D_S-1}\Pr\{X_{n+1}=m, D_k=j, D_{k+1}=j', D_{k+2}=l_1, D_{k+3}=l_2\}$$

$$+ \cdots\cdots$$

$$+\sum_{j=0}^{D_S-1}\sum_{j'=D_S}^{D_{max}}\sum_{l_1=D_S}^{D_{max}}\sum_{l_2=D_S}^{D_{max}}\cdots\sum_{l_{\Gamma-2}=D_S}^{D_{max}}\sum_{l_{\Gamma-1}=0}^{D_S-1}\Pr\{X_{n+1}=m, D_k=j, D_{k+1}=j', D_{k+2}=l_1,$$

$$D_{k+3}=l_2,\ldots, D_{k+\Gamma-1}=l_{\Gamma-2}, D_{k+\Gamma}=l_{\Gamma-1}\} \quad . \qquad (6\text{-}3).$$

In eqn. (6-3), $\Pr\{X_{n+1} = m\}$ is computed as the sum of Γ entries. The first entry computes the probability of the case that (X_{n+1} = m) and the $(k+1)$st arrival passes the rate-controller successfully, thereby becoming the (n+1)st departure. The second entry computes the probability of the case that (X_{n+1} = m), the $(k+1)$st arrival is discarded, and the $(k+2)$nd arrival passes the rate-controller, thereby becoming the (n+1)st departure. The third entry computes the probability of the case that (X_{k+1} = m), both the $(k+1)$st and the (k+2)nd arrivals are discarded, and the $(k+3)$rd arrival passes the rate-controller, thereby becoming the $(n+1)$st departure. The Γth entry computes the probability of the case that (X_{k+1} = m), $(\Gamma-1)$ cells are discarded consecutively from the $(k+1)$st until the $(k+\Gamma-1)$th arrival, and the $(k+\Gamma)$th arrival passes successfully, thereby becoming the $(n+1)$st departure.

Obviously, if the probability distribution of the inter-departure time were calculated exactly by using eqn. (6-3), the calculation would be very complex. However, the cell loss ratio allowed in high speed networks is usually very small. For example, the cell loss ratio that can be tolerated by video traffic is around 10^{-9} - 10^{-7}, and the voice traffic can tolerate a cell loss ratio of 10^{-4} -10^{-5}. The cell loss probability allowed at a rate-controller must be smaller than these tolerable bounds. Therefore, the probability of having consecutive cell losses is extremely small.

So, the inter-departure time distribution can be approximately computed by using only the first one or two item(s) in eqn. (6-3).

The first entry in eqn. (6-3) can be computed as follows:

$$\sum_{j=0}^{D_S-1}\sum_{j'=0}^{D_S-1}\Pr\{X_{n+1}=m, D_k=j, D_{k+1}=j'\}$$

$$=\sum_{j=0}^{D_S-1}\sum_{j'=0}^{D_S-1}\sum_{i=0}^{\Lambda-\Gamma}\sum_{h'=A_{\min}}^{A_{\max}}\sum_{i'=0}^{\Lambda-\Gamma}\{p_1(j,i,m+j-j';j',i',h')\cdot\pi(j,i)\cdot a(m+j-j')\} \quad .$$

The second entry in eqn. (6-3) can be computed as follows:

$$\sum_{j=0}^{D_S-1}\sum_{j'=D_S}^{D_{\max}}\sum_{l_1=0}^{D_S-1}\Pr\{X_{n+1}=m, D_k=j, D_{k+1}=j', D_{k+2}=l_1\}$$

$$=\sum_{j=0}^{D_S-1}\sum_{j'=D_S}^{D_{\max}}\sum_{l_1=0}^{D_S-1}\sum_{h_1=A_{\min}}^{A_{\max}}\sum_{i=0}^{\Lambda-\Gamma}\sum_{i'=0}^{\Lambda-2\Gamma+T}\sum_{s_1=0}^{\Lambda-\Gamma}\{p_1(j',i',j'-l_1;l_1,s_1,h_1)$$

$$\cdot p_1(j,i,j+m-j';j',i',j'-l_1)\cdot\pi(j,i)\cdot a(j+m-j')\} \quad ,$$

where $p_1(j,i,h;j',i',h')$ is the transitional probability of the Markov Chain $\{D_k, S_k, A_k\}_{k\geq 1}$, which is obtained as follows:

$$p_1(j,i,h;j',i',h')=\begin{cases} a(h') & \text{if (condition of case (1)) and (((i' < } \Lambda - \Gamma\text{) and (h = i' -i + j + } \Gamma\text{))} \\ & \quad \text{or ((i' = } \Lambda - \Gamma\text{) and (h} \geq \Lambda + \text{j - i)))} \\ a(h') & \text{if (condition of case (2)) and (h = T + j - j')} \\ a(h') & \text{if (condition of case (3)) and (h = } \Gamma + \text{j - j' -i)} \\ a(h') & \text{if (condition of case (4)) and (((i' < } \Lambda - \Gamma\text{) and (h = j + i' -i))} \\ & \quad \text{or ((i' = } \Lambda - \Gamma\text{) and (h} \geq \Lambda + \text{j - i - } \Gamma\text{)))} \\ a(h') & \text{if (condition of case (5)) and (h = j - j')} \\ 0 & \text{otherwise.} \end{cases}$$

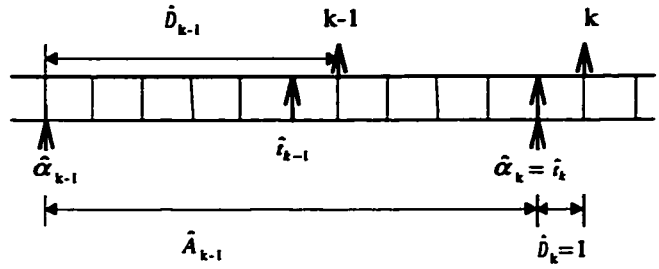
The condition of case (1, 2, ..., 5) is the same as the condition of case (1, 2, ..., 5) in section 6.2.2, when we derived the value of *p(j, i; j', i')*.

6.3 Queuing Analysis of the Multiplexer

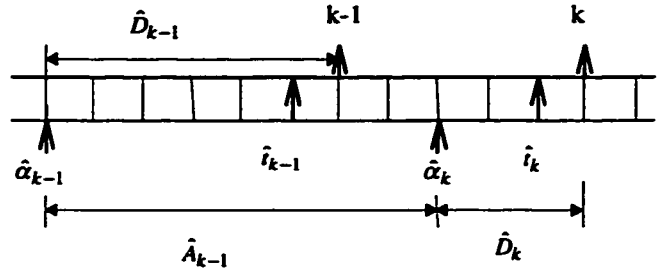
Under the assumption of generally distributed inter-arrival time process for the reference stream, we analyze in this section the statistical performances for the reference stream at a FCFS multiplexer, including the cell delay distribution and the inter-departure time process. In order to simplify the analysis, we make two additional assumptions. First we assume that, under the FCFS discipline, the reference stream has the so-called "mini-priority", i.e., the reference cells are served before those background cells that arrive during the same slot. Second, we assume that the multiplexing buffer is big enough so that no cell loss will be incurred due to an overflow in the buffer.

6.3.1 Cell delay distribution

Based on our network model in Fig. 6-1, the cell delay at the multiplexer consists of the queuing delay at the output buffer and the cell transmission time. Let $\hat{\alpha}_k$ be the arrival time of the kth cell at the multiplexer, which is the end of the slot over which the kth cell is completely received by the multiplexer. Let $\hat{t}_k$ be the service epoch of the kth cell, defined as the beginning of the slot over which the kth cell is transmitted. Then $\hat{D}_k = \hat{t}_k - \hat{\alpha}_k + 1$ is the delay of the kth cell at the multiplexer. Clearly, the process $\{\hat{D}_k\}_{k\geq 1}$ forms a Markov chain embedded at the service epoch $\{\hat{t}_k\}_{k\geq 1}$, because the knowledge of the value of $\hat{D}_k$ alone is sufficient to determine the probability distribution of $\hat{D}_{k+1}$. With reference to section 2.2.3, $\hat{A}_k$ is the inter-arrival time between the kth and the $(k+1)$st arrival from the reference stream at the multiplexer, the maximum (minimum) value of $\hat{A}_k$ is $\hat{A}_{max}$ ($\hat{A}_{min}$), and the probability distribution of $\hat{A}_k$ is $\hat{a}(n) = \Pr\{\hat{A}_k = n\}$. We let $Q_b(t)$ denote the number of background cells in the multiplexing buffer at time t.



(a) When ($\hat{D}_{k-1} < \hat{A}_{k-1}$) and ($Q_b(\hat{\alpha}_k - 1)=0$)



(b) When ($\hat{D}_{k-1} < \hat{A}_{k-1}$) and ($Q_b(\hat{\alpha}_k - 1) > 0$)

Fig. 6-5 Delay Analysis When ($\hat{D}_{k-1} < \hat{A}_{k-1}$)

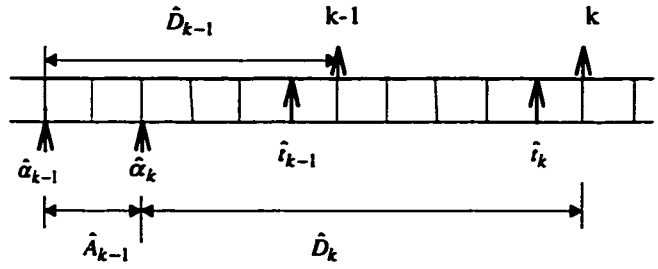


Fig. 6-6 Delay Analysis When ($\hat{D}_{k-1} \geq \hat{A}_{k-1}$)

With reference to Fig. 6-5 when $\hat{D}_{k-1} < \hat{A}_{k-1}$, and with reference to Fig. 6-6 when ($\hat{D}_{k-1} \geq \hat{A}_{k-1}$), we derived the expressions for $\hat{D}_k$ as follows:

$$\hat{D}_k = \begin{cases} 1 & \text{if } (1 \le \hat{D}_{k-1} < \hat{A}_{k-1}) AND (Q_b(\hat{\alpha}_k - 1) = 0) \\ Q_b(\hat{\alpha}_k - 1) & \text{if } (1 \le \hat{D}_{k-1} < \hat{A}_{k-1}) AND (Q_b(\hat{\alpha}_k - 1) > 0) \\ \hat{D}_{k-1} - \hat{A}_{k-1} + \Psi_{\hat{A}_{k-1}} + 1 & \text{if } (\hat{D}_{k-1} \ge \hat{A}_{k-1}) \end{cases} \quad (6\text{-}4)$$

The above eqn. (6-4) indicates that there are the following three cases when we computing the value of $\hat{D}_k$.

In case (1), we have ($\hat{D}_{k-1} < \hat{A}_{k-1}$) and ($Q_b(\hat{\alpha}_k\text{-}1) = 0$), where $Q_b(\hat{\alpha}_k\text{-}1)$ is the number of background cells in the multiplexing buffer at time ($\hat{\alpha}_k$-1). Based on the "mini-priority" assumption, the background cells arriving after time ($\hat{\alpha}_k$-1) will be served after the *k*th cell in the reference stream. The inequality ($\hat{D}_{k-1} < \hat{A}_{k-1}$) means that the *(k-1)*st cell has been served before the arrival of the *k*th cell. The inequality ($Q_b(\hat{\alpha}_k - 1) = 0$) means that no background cell will be served before *k*th reference cell. Therefore, the *k*th cell will be served right after its arrival, and its delay is equal to its transmission time of *1* slot.

In case (2), we have ($\hat{D}_{k-1} < \hat{A}_{k-1}$) and ($Q_b(\hat{\alpha}_k\text{-}1) > 0$). The inequality ($\hat{D}_{k-1} < \hat{A}_{k-1}$) means that the *(k-1)*st cell has been served before the arrival of the *k*th cell. The inequality ($Q_b(\hat{\alpha}_k\text{-}1) > 0$) means that the server will start to serve the *k*th reference cell only after it finishes serving $Q_b(\hat{\alpha}_k\text{-}1)$ background cells. Since $Q_b(\hat{\alpha}_k\text{-}1)$ cells include the one which is in service at time ($\hat{\alpha}_k$-1), the delay of the *k*th reference cell will be $Q_b(\hat{\alpha}_k\text{-}1)$.

In case (3), we have ($\hat{D}_{k-1} \ge \hat{A}_{k-1}$). This inequality mean that the *k*th cell arrives before the *(k-1)*st cell is transmitted. In this case, the *k*th reference cell will first wait for ($\hat{D}_{k-1} - \hat{A}_{k-1}$) slots until the *(k-1)*st reference cell is transmitted. After that, it has to wait until the server finishes serving $\Psi_{\hat{A}_{k-1}}$ background cells that arrive between $\hat{\alpha}_{k-1}$ and ($\hat{\alpha}_k$ -1). Thus, the delay of the *k*th cell is $\hat{D}_{k-1} - \hat{A}_{k-1} + \Psi_{\hat{A}_{k-1}} + 1$.

Let $\psi_n(j)$ = the probability of j background arrivals during any interval of n slots;

$P_{q,q'}^{(l)}$ = the l-step transitional probability for Geo / D / 1 queue;

$\hat{p}(j;j')$ = the transitional probability of $\{\hat{D}_k\}_{k\geq 1}$ from $\hat{D}_{k-1}=j$ to $\hat{D}_k=j'$;

$A_1 = \max(j+1, \hat{A}_{\min})$;

$A_2 = \min(j, \hat{A}_{\max})$;

$N_1 = \min(n \bullet \Psi_1, j' - j - 1 + n)$.

Based on the discussions of the three cases in eqn.(6-4), the transitional probabilities $\hat{p}(j;j')$ can be derived as follows:

1) When $1 \leq j \leq \hat{A}_{\max} - 1$

$$\hat{p}(j;j') = \sum_{n=A_1}^{\hat{A}_{\max}} \sum_{k=0}^{\Psi_1(j+1)} \hat{a}(n) \cdot \psi_{j+1}(k) \cdot [p_{k,0}^{(n-j-1)} \cdot 1\{j'=1\} + p_{k,j'}^{(n-j-1)}]$$
$$+ \sum_{n=\hat{A}_{\min}}^{A_2} \sum_{k=0}^{N_1} \hat{a}(n) \cdot \psi_n(k) \cdot 1\{j' = j-n+k+1\}$$

2) When $j \geq \hat{A}_{\max}$

$$\hat{p}(j;j') = \sum_{n=\hat{A}_{\min}}^{A_2} \sum_{k=0}^{N_1} \hat{a}(n) \cdot \psi_n(k) \cdot 1\{j' = j-n+k+1\}.$$

3) Otherwise, $\hat{p}(j;j') = 0$.

The l-step transitional probability $P_{q,q'}^{(l)}$ for Geo / D / 1 queue is obtained by using the i. i. d. batch arrival model of the background traffic (section 2.2.4). When only the background traffic is served during an arbitrary interval $[h, h+l]$, the number of cells that arrive before time h and are still in the queue at time $(h+l)$ evolves as the queue length process of the Geo/D/1 queue [LaSt97]. Therefore, we can write

$$P_{q,q'}^{(1)} = \psi_1(q'-q+1) + \psi_1(0) \cdot 1(q+q'=0) \quad ;$$

$$P^{(l)}_{q,q'} = \sum_r P^{(l-1)}_{q,r} \cdot P^{(1)}_{r,q'} \quad .$$

Thus, the transitional probability matrix $\hat{P}$ of the Markov Chain $\{ \hat{D}_k \}_{k \geq 1}$ can be obtained by assigning the value of $\hat{p}(j;j')$ to a corresponding matrix element with j and j' being its row and column indices. Hence, the transitional probability matrix is

$$\hat{P} = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \cdots \\ A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & A_0 & A_1 & A_2 & \cdots \\ 0 & 0 & A_0 & A_1 & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdots \end{bmatrix}$$

For $v \geq 0$, B_v and A_v are $(\hat{A}_{max}\text{-}1)$ by $(\hat{A}_{max}\text{-}1)$ matrices. Note that $\hat{P}$ is similar to the one for an ordinary M/G/1 queue with its blocks reduced to scalars. Let $\bar{\hat{\pi}}$ denote the equilibrium probability vector for this Markov process, i.e., $\bar{\hat{\pi}} = \bar{\hat{\pi}} \bullet \hat{P}$. The vector $\bar{\hat{\pi}}$ can be solved by using the method developed by Neuts in [Neut89]. Without repeating any proofs, we summarize the steps to solve $\bar{\hat{\pi}}$ from $\bar{\hat{\pi}} = \bar{\hat{\pi}} \bullet \hat{P}$ in Appendix D.2. By following those steps, we obtain the stationary cell delay distribution at a FCFS multiplexer as:

$$\Pr\{\text{cell delay} = j\} = \hat{\pi}[j] \quad .$$

6.3.2 The inter-departure process

In this sub-section, we derive the distribution of the inter-departure time between two successive departure cells from the multiplexer. The Markov chain $\{ \hat{D}_k \}_{k \geq 1}$ does not provide enough information to compute the distribution of inter-departure time. Therefore we need to describe the system using the Markov chain $\{ \hat{D}_k, \hat{A}_k \}_{k \geq 1}$, where $\hat{A}_k = \hat{\alpha}_{k+1} - \hat{\alpha}_k$ indicates the inter-arrival time between the kth and the $(k+1)$st arriving cell from the reference stream. Let $\hat{\pi}_1(j,h)$ denote

the equilibrium probability for this Markov chain. From eqn. (6-4), we can see that $\hat{D}_k$ is independent of $\hat{A}_k$. Thus, the equilibrium probability for the Markov chain $\{\hat{D}_k, \hat{A}_k\}_{k\geq 1}$, can be computed as $\hat{\pi}_1(j,h) = \hat{\pi}(j) \cdot \hat{a}(h)$.

Let $\hat{p}_1(j,h;j',h')$ = transitional probability of the Markov Chain $\{\hat{D}_k, \hat{A}_k\}_{k\geq 1}$ from $\hat{D}_{k-1}$= j *and* $\hat{A}_{k-1} = h$ *to* $\hat{D}_k = j'$ *and* $\hat{A}_k = h'$;

$A_1 = \max(j+1, \hat{A}_k)$;

$N_1 = \min(n \bullet \Psi_1, j' - j - 1 + n)$.

Based on the above discussions and eqn. (6-4), $\hat{p}_1(j,h;j',h')$ can be derived into the following two cases:

1) When $1 \leq j \leq \hat{A}_{\max} - 1$

$$\hat{p}_1(j,h;j',h') = \sum_{k=0}^{\Psi_1(j+1)} \psi_{j+1}(k) \cdot [p_{k,0}^{(h-j-1)} \cdot 1\{j'=1\} + p_{k,j'}^{(h-j-1)}] \cdot 1\{A_1 \leq h \leq \hat{A}_{\max}\} \cdot \hat{a}(h') + \sum_{k=0}^{N_1} \psi_h(k) \cdot 1\{j' = j-h+k+1\} \cdot 1\{\hat{A}_{\min} \leq h \leq \hat{A}_{\max}\} \cdot \hat{a}(h') \quad .$$

2) When $j \geq \hat{A}_{\max}$

$$\hat{p}_1(j,h;j',h') = \sum_{k=0}^{N_1} \psi_h(k) \cdot 1\{j' = j-h+k+1\} \cdot 1\{\hat{A}_{\min} \leq h \leq \hat{A}_{\max}\} \cdot \hat{a}(h') \quad .$$

Let $\hat{X}_{k+1}$ be the inter-departure time between the kth departure and the $(k+1)$st departure cell. Then, we have $\hat{X}_{k+1} = \hat{D}_{k+1} + \hat{A}_k - \hat{D}_k$. The distribution of the inter-departure time is derived as follows:

$$\Pr\{X_{k+1} = x\} = \sum_{j=1}^{\infty} \sum_{h=\hat{A}_{\min}}^{\hat{A}_{\max}} \sum_{h'=\hat{A}_{\min}}^{\hat{A}_{\max}} \hat{p}_1(j,h;x+j-h,h') \cdot \hat{\pi}(j,h) \quad , \qquad (6\text{-}5)$$

where

$$A' = min\{j+x-1,\ \hat{A}_{max}\} \quad .$$

The results obtained so far in both the previous and the current sections will be used in the next section on the end-to-end performance analysis.

6.4 End-to-end Performance Analysis

In this section, the statistical end-to-end performance will be analyzed for the reference stream. We will use the results obtained in both sections 6.2 and 6.3 to examine how the characteristics of the reference stream changes as it flows from source to destination, and to derive its end-to-end delay distribution.

We study a tandem network consisting of M nodes (see Fig. 6-1). There are $(N_b + 1)$ traffic streams arriving at each node: the reference stream and N_b background traffic streams. The FCFS policy is applied at each multiplexer, and our proposed Leaky-Bucket mechanism is used at each rate-controller. In order to simplify analysis, we make the following three assumptions. First, we assume that only the reference stream traverses from the source node 1 to the destination node M, while all the background streams leave the network immediately after it departing a node. Second, we assume that the inter-departure process from each multiplexer and rate-controller is an i. i. d. process. This assumption enables us to investigate how the statistical characteristics of a reference stream will vary as it passes through multiple nodes. That is, the reference stream's inter-departure time process from one network component can be fed into the next network component on the path as the inter-arrival time process. Third, we assume that the cell delay at one multiplexer or rate-controller is independent of the delay at any other multiplexer or rate-controller.

We want to investigate the end-to-end performance of the reference stream in two types of tandem network: (1) The RCS network (Fig. 6-1) where each node consists of one rate-controller and one multiplexer, and the reference stream traverses the rate-controller before entering the multiplexer. (2) The conventional tandem network, in which each node only consists of a traffic multiplexer, i.e., there is no rate-controller in the conventional network model. We compare the performance in these two types of network, in order to study the impact of rate-controlling on the end-to-end QoS provisioning.

6.4.1 The change of traffic characteristics

We study how the inter-cell time distribution of the reference stream changes as it flows from source to destination. Let $\{\hat{X}_m\}_{1\le m\le M}$ be the inter-departure time process of the reference stream from the multiplexer of the mth node. Its probability distribution is denoted by $\hat{x}^{(m)}(n) = \Pr\{\hat{X}_m = n\}$. Let $\{\hat{A}_m\}_{1\le m\le M}$ be the inter-arrival time process of the reference stream at the multiplexer of the mth node. Its probability distribution is denoted by $\hat{a}^{(m)}(n) = \Pr\{\hat{A}_m = n\}$. Let $\{X_m\}_{1\le m\le M}$ be the inter-departure time process of the reference stream from the rate-controller at the mth node. Its probability distribution is denoted by $x^{(m)}(n) = \Pr\{X_m = n\}$. Let $\{A_m\}_{1\le m\le M}$ be the inter-arrival time process of the reference stream at the rate-controller of the mth node. Its probability distribution is denoted by $a^{(m)}(n) = \Pr\{A_m = n\}$. In the RCS network, with the assumption of an i. i. d. inter-departure time process from the rate-controller and multiplexer, we can use the process $\hat{X}_m$ ($1 \le m \le M-1$) to approximate A_{m+1} ($1 \le m \le M-1$) the inter-arrival time process of the reference stream at the rate-controller of the $(m+1)$st node, and use the process X_m ($1 \le m \le M$)

to approximate $\hat{A}_m$ ($1 \le m \le M$) the inter-arrival time process of the reference stream at the mth multiplexer. That is,

$$a^{(m+1)}(n) = \hat{x}^{(m)}(n) \qquad \text{for } 1 \le m \le M\text{-}1 \ .$$

$$\hat{a}^{(m)}(n) = x^{(m)}(n) \qquad \text{for } 1 \le m \le M \ .$$

To investigate the change of traffic characteristics for the reference stream along its path in the RCS network, we reiterate the distribution of inter-departure process over a series of multiplexers and rate-controllers, as described in the following steps.

Step 1: Initialize m=1.

Step 2: Feed $\{A_m\}$ as the inter-arrival process of the rate-controller at node m. Obtain $x^{(m)}(n)$ the distribution function of $\{X_m\}$ by using eqn. (6-3).

Step 3: Feed $\{X_m\}$ as $\{\hat{A}_m\}$ the inter-arrival process of the multiplexer at node m. Obtain $\hat{x}^{(m)}(n)$ the distribution function of $\{\hat{X}_m\}$ by using eqn. (6-5).

Step 4: Feed $\{\hat{X}_m\}$ as $\{A_{m+1}\}$ the inter-arrival process of the rate-controller at node (m+1), i.e., $a^{(m+1)}(n) = \hat{x}^{(m)}(n)$. Obtain $x^{(m+1)}(n)$ the distribution function of $\{X_{m+1}\}$ by using the eqn. (6-3).

Step 5: Feed $\{X_{m+1}\}$ as $\{\hat{A}_{m+1}\}$ the inter-arrival process of the multiplexer at the $(m+1)$st node, i.e., $\hat{a}^{(m+1)}(n) = x^{(m+1)}(n)$. Obtain $\hat{x}^{(m+1)}(n)$ the distribution function of $\{\hat{X}_{m+1}\}$ by using eqn. (6-5).

Step 6: Let m=m+1. If $m \le M - 1$, go to step 4. Otherwise, stop.

Note that the inter-arrival time processes $\{A_m\}$ and $\{\hat{A}_m\}$ of the reference stream must be bounded in order for the numerical computation to be applicable. That is, $A_{m,\,min} \le A_m \le A_{m,\,max}$ and $\hat{A}_{m,\min} \le \hat{A}_m \le \hat{A}_{m,\max}$ for $1 \le m \le M$. Therefore, truncation operation is needed in order to avoid high computational cost. The truncation of $A_{m,\,max}$ is conducted by letting $A_{m,\,max} = \max\{n: \Pr(X_{m-1} = n) \ge \varepsilon\}$, where $\varepsilon \ge 0$. Obviously, there is inaccuracy caused by truncation when $\varepsilon > 0$. As the value of ε decreases, the accuracy of numerical results will be improved at the price of

higher computational complexity. When $\varepsilon = 0$, there is no truncation, but the computational cost for $\{X_m\}$ and $\{\hat{X}_m\}$ could be very high due to the possible large values of $A_{m,max}$ and $\hat{A}_{m,\max}$.

We use a similar approach in the conventional network. That is, we let

$$\hat{a}^{(m+1)}(n) = \hat{x}^{(m)}(n) \qquad \text{for } 1 \le m \le M-1 .$$

To study the change of traffic characteristics for the reference stream along its path in the conventional network, we reiterate the distribution of inter-departure process over a series of multiplexers, as described in the following steps.

Step 1: Initialize m=1.

Step 2: Feed $\{\hat{A}_m\}$ as the inter-arrival process of the multiplexer at node m. Obtain $\hat{x}^{(m)}(n)$ the distribution function of $\{\hat{X}_m\}$ by using eqn. (6-5).

Step 3: Feed $\{\hat{X}_m\}$ as $\{\hat{A}_{m+1}\}$ the inter-arrival process of the multiplexer at node (m+1), i.e., $a^{(m+1)}(n) = \hat{x}^{(m)}(n)$. Obtain $\hat{x}^{(m+1)}(n)$ the distribution function of $\{\hat{X}_{m+1}\}$ by using eqn. (6-5).

Step 4: Let m=m+1. If $m \le M-1$, go to step 3. Otherwise, stop.

6.4.2 End-to-end delay distribution

We study the probability distribution of the end-to-end delay for the reference stream. Let $\{\hat{D}_m\}_{1 \le m \le M}$ be the queuing delay process of the reference stream at the mth multiplexer. Its probability distribution is denoted by $\hat{d}^{(m)}(n) = \Pr\{\hat{D}_m = n\}$. Let $\{D_m\}_{1 \le m \le M}$ be delay process of the reference stream at the rate-controller of the mth node. Its probability distribution is denoted by $d^{(m)}(n) = Pr\{D_m = n\}$. The queuing delay distribution at the rate-controller and that at the multiplexer have been derived in sections 6.2.2 and 6.3.1, respectively. Using the independence

assumption in section 6.4, the end-to-end delay distribution $eted(n) = Pr\{\text{end-to-end delay} = n \text{ slots}\}$ of the RCS network can be obtained as:

$$eted(n) = (d^{(1)}(n) \otimes \hat{d}^{(1)}(n)) \otimes (d^{(2)}(n) \otimes \hat{d}^{(2)}(n)) \otimes \ldots \otimes (d^{(M)}(n) \otimes \hat{d}^{(M)}(n)) \quad .$$

Similarly in the conventional network, the end-to-end delay distribution can be obtained as:

$$eted(n) = \hat{d}^{(1)}(n) \otimes \hat{d}^{(2)}(n) \otimes \ldots \otimes \hat{d}^{(M)}(n) \quad .$$

The convolution can be computed efficiently by using the Fast Fourier Transform (FFT). The detail numerical method can be found in [PFTV86].

6.5 Performance Evaluation

In this section, we run simulations to obtain the end-to-end performance in a tandem ATM network. Three end-to-end QoS measures are evaluated for an individual VBR connection: the inter-departure time process, the end-to-end delay distribution, and the mean end-to-end delay. By comparing the end-to-end performance in an RCS network with that in a conventional network, we investigate the impact of rate-controlling on the end-to-end QoS provisioning.

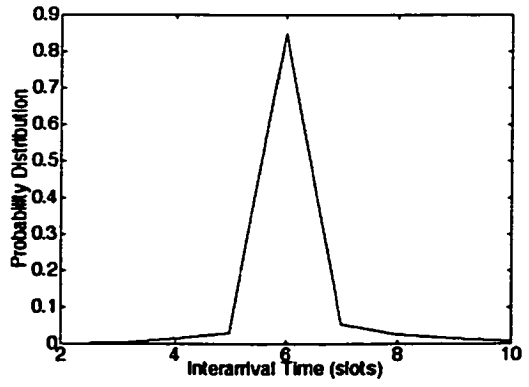


Fig. 6-7 The Inter-arrival Time Process of Reference Stream

In this section, the number of nodes in the tandem network is M = 5. The inter-arrival time process of the reference stream is illustrated in Fig. 6-7, from which we obtain that the average rate of the reference stream is 0.17 cells/slot. The number of the background streams is N_b = 10. The traffic load of each background stream is p = 0.053. Then, the total traffic load in the network comes to 0.7 cells/slot.

We ran simulations in three different cases. In the first case, ten background traffic streams are input into each node, and they traverse only one hop, i.e., they leave the tandem network immediately after departing from a node. In the second case, two out of ten background streams input at node *1* traverse to the destination node *5*. The other eight background streams input at node *1* traverse only one hop. Eight new background streams are input at each intermediate node and the destination node, and they also traverse only one hop. In the third case, eight out of ten background streams input at node *1* traverse to the destination. The other two background streams input at node *1* only traverse one hop. Two new background streams are input at each intermediate node and the destination node, and they also traverse only one hop.

The rationale for selecting the above three cases is: the first case implies a large-scale network with a number of independent sessions in it. Thus, it is reasonable to assume that most of the background streams have different paths from that of the reference stream. The third case implies a small-scale network with many sessions having the same source and destination. Thus, it is reasonable to assume that many background streams have the same path as that of the reference stream. The second case implies a medium-scale network.

The simulation is conducted using the simulation tool OPNET. Each simulation was run for 5,000,000 time units. The number of packets involved in one simulation is 3,500,000.

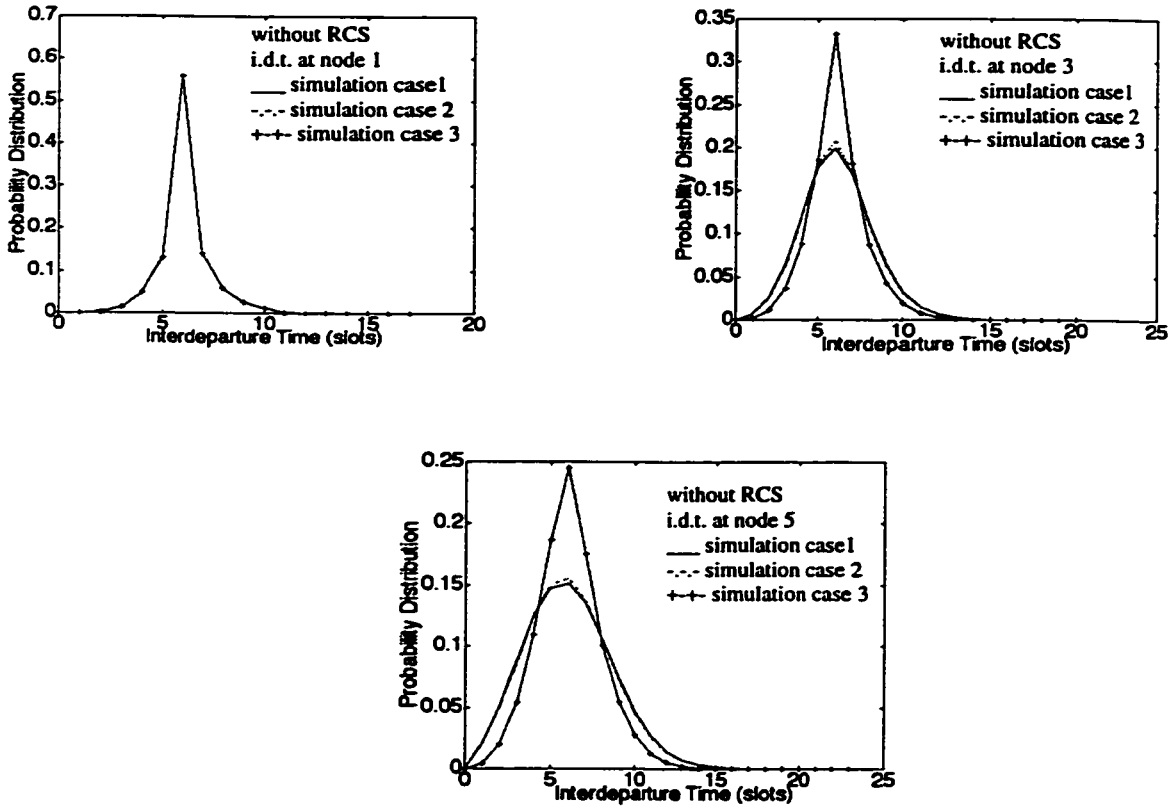


Fig. 6-8 Distribution of Inter-departure Time at Node 1, 3, 5 (Without RCS and the total load is 0.7 cells/slot)

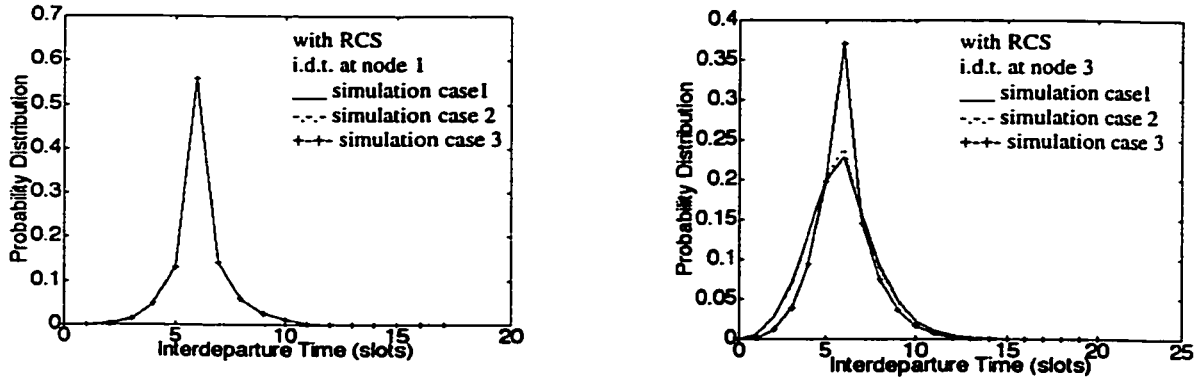


Fig. 6-9 Distribution of Inter-departure Time at Node 1, 3, 5 (to be continued) (With RCS and the total load is 0.7 cells/slot)

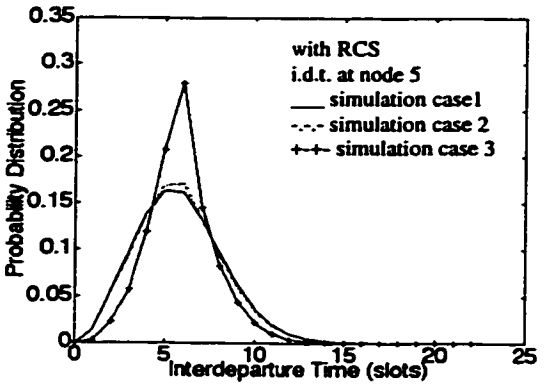


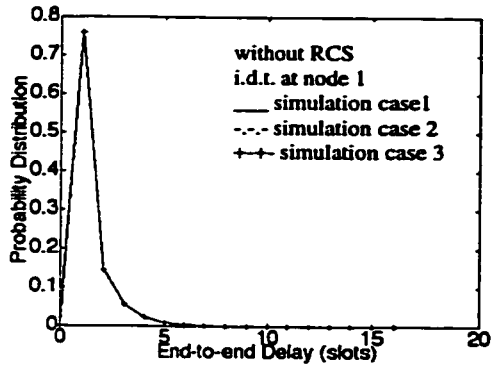
Fig. 6-9 Distribution of inter-departure time at node 1, 3, 5 (With RCS and the total load is 0.7 cells/slot)

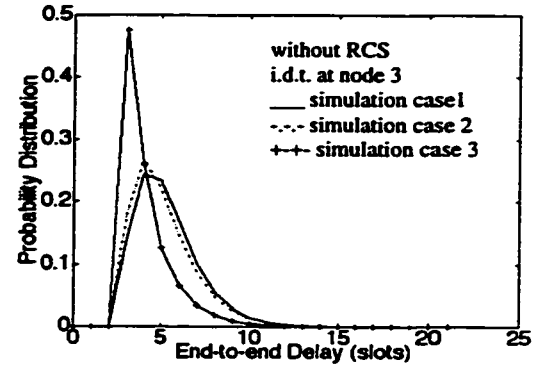
First, we investigate how the traffic characteristics of the reference stream changes along the path, i.e., we look at the inter-departure time process. Figures 6-8 and 6-9 show the probability distribution of the interdeparture time of the reference stream at node 1, 3, and 5. Figures 6-8 shows the performance in a network without rate-controlling. Figures 6-9 shows the performance in an RCS network. We observe:

(1) The probability distribution of the i. d. t. (inter-departure time) of the reference stream flattens out as the stream traversing more nodes.

(2) The probability distribution of the i. d. t. at node 5 when the RCS discipline is used is more sharpened than the corresponding probability distribution when the RCS is not used. This means that the RCS discipline helps to keep the traffic characteristics and reduce the distortion induced by multiplexing.

(3) The probability distribution of i. d. t. in the simulation case 3 is more sharpened than that in case 1 or 2. This implies that the reference stream tends to keep its characteristics in a small-scaled network, where many sessions have the same source and destination.





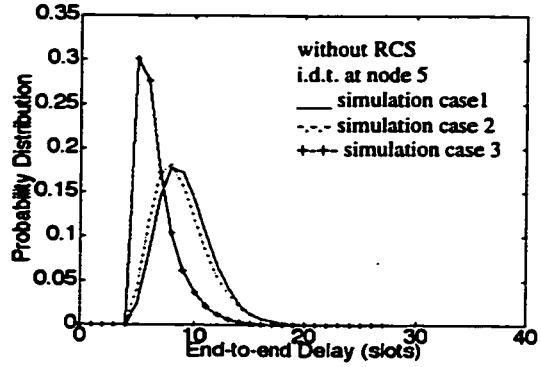
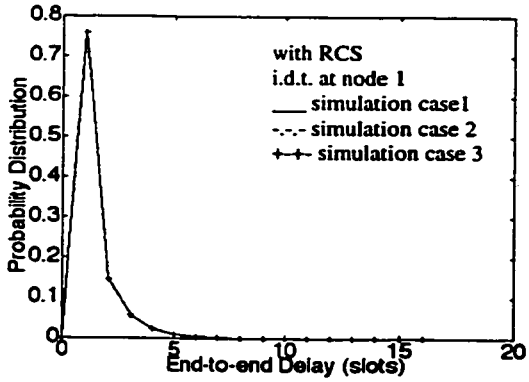


Fig. 6-10 Distribution of End-to-End Delay at Node 1, 3, 5 (Without RCS and the total load is 0.7 cells/slot)



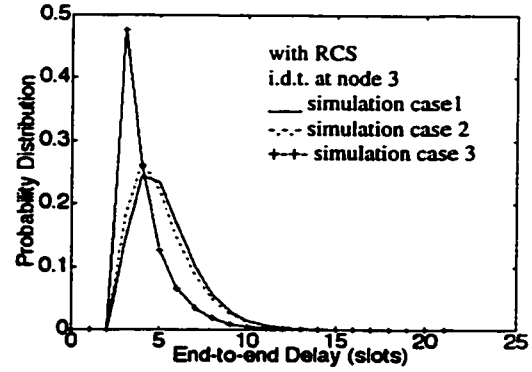


Fig. 6-11 Distribution of End-to-End Delay at Node 1, 3, 5 (to be continued) (With RCS and the total load is 0.7 cells/slot)

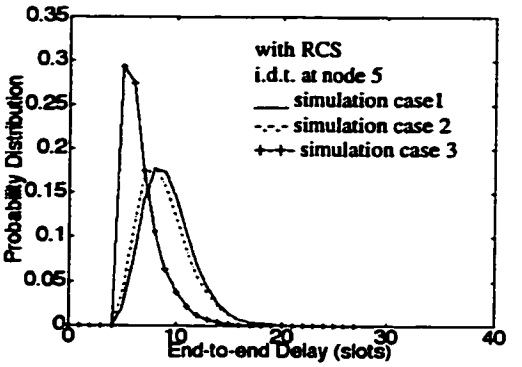


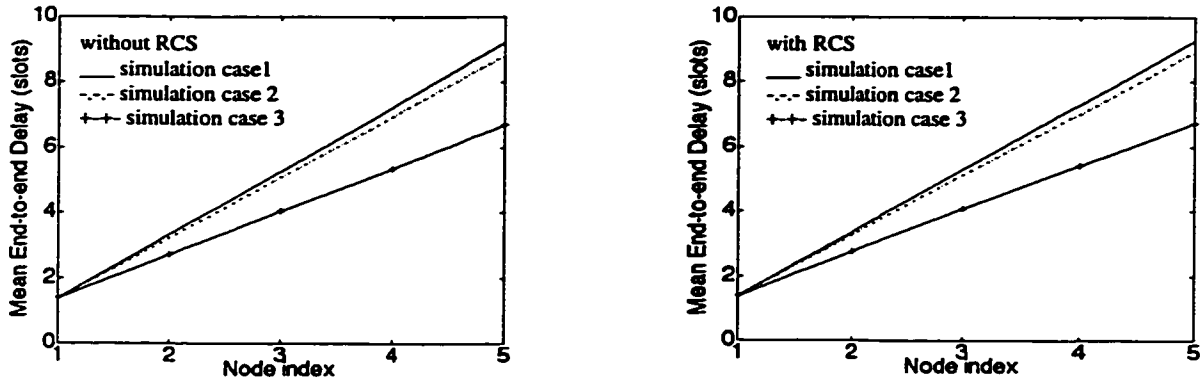
Fig. 6-11 Distribution of End-to-End Delay at Node 1, 3, 5 (With RCS and the total load is 0.7 cells/slot)

Next, we look at the end-to-end delay performance. Figures 6-10 and 6-11 show the end-to-end delay distribution of the reference stream at node 1, 3, and 5. Figures 6-10 shows the performance in a network without rate-controlling. Figures 6-11 shows the performance in an RCS network. We observe:

(1) The end-to-end delay distribution region moves to the right with the increasing number of hops, which means more cells have longer delay after traversing more hops, as it should be.

(2) Comparing Fig. 6-10 with 6-11, we see that more cells have slightly longer end-to-end delay at node 3 and 5 when RCS discipline is used than that when RCS discipline is not used. That is, less distorted traffic is output from these nodes at the price of increased end-to-end delay.

(3) Most cells of the reference stream have smaller end-to-end delay in case 3 than in case 1 or 2. This means that, when traversing the same number of nodes, the reference stream has shorter end-to-end delay in a small-scaled network than that in a large-scaled network.



(a) without RCS (b) with RCS

Fig. 6-12 Mean End-to-End Delay

The mean end-to-end delay performance is presented in Fig. 6-12. We observe:

(1) The mean end-to-end delay in simulation case 2 is slightly shorter than that in simulation case 1. The mean end-to-end delay in case 3 is much shorter than that in both cases 1 and 2. Thus, when traversing the same number of nodes, the mean end-to-end delay of the reference connection in a small-scaled network is much shorter than that in a medium-scaled or large-scaled network.

(2) The mean end-to-end delay in the RCS network is only slightly larger than that in the conventional network.

6.6 Chapter Summary and Remarks

In this chapter, we investigated the impact of rate-controlling on end-to-end statistical QoS provisioning, through both analysis and simulation. First, we proposed a new operating mechanism for the leaky bucket rate-controller, and analyzed the queuing performance for the rate-controller. Second, we studied the queuing performance at a FCFS multiplexer. Third, the

end-to-end performances of an individual connection were obtained. Finally, simulations were conducted to investigate the impact of rate-controlling on the end-to-end performance.

Chapter 7

Design Guidelines

7.1 Introduction

ATM is a connection-oriented communication protocol. When provisioning an end-to-end VBR traffic connection, we must consider the design of its traffic control strategies. There are three major control strategies for VBR traffic: connection admission control, traffic shaping, and statistical multiplexing.

First, every network node must have the connection admission control function in order to prevent network congestion and QoS degradation. So in chapter 3, an efficient admission control algorithm was proposed for the real-time VBR traffic. The guideline on how to use it is provided in the following section 7.2.

Second, the traffic shaper has become an essential traffic control element in an ATM network. So in Chapter 4, two minimum MLB traffic shapers guaranteeing the specified shaping delay bound or the specified shaping buffer occupancy were presented. The guideline on how to apply our proposed shaper design is given in section 7.3.

Third, for the purpose of achieving high resource utilization, statistical multiplexing must be used when provisioning VBR traffic due to the burstiness of VBR traffic. So in Chapter 5, the statistical traffic shaping scheme was put forward. The guideline on how to apply it is provided in section 7.4.

Fourth, the ultimate goal of VBR traffic control is to support end-to-end QoS. So in Chapter 6, the end-to-end performance was analyzed for an individual VBR traffic flow. The guideline on how to obtain the end-to-end performance is provided in section 7.5.

7.2 Design Guideline for Connection Admission Control

Connection admission control is a necessary component of the end-to-end VBR traffic control. It is used in the connection setup phase to accept or reject an incoming connection based on the available resources and the requested attributes.

Because ATM connections are generally provisioned node-by-node, admission control should desirably be distributive. That is, the same software version implementing CAC (Connection Admission Control) is installed at every node. When activated by the connection Request message, the CAC software checks if the new connection's QoS requirements can be satisfied without degrading the QoS of the existing connections. This condition is verified along the path of the new connection (i.e., at the source node, the intermediate nodes, and the destination node).

To design an admission control algorithm, we need to know the network environment. Our design assumes an RCS network (section 2.1), where the leaky bucket mechanism is used at every rate-controller, and the EDF scheduling policy is used at every traffic multiplexer. We need to choose the traffic model to specify the characteristics of an individual VBR traffic flow, and to decide on the VBR connection's QoS requirements. In our algorithm, the MLB traffic model is used to characterize a VBR traffic flow, and the end-to-end delay bound is selected as the QoS requirement. The deterministic service is supported for each VBR connection, i.e., no cell loss is guaranteed. In addition, we need to decide the key network resources to be considered in the admission control. In our algorithm, the link capacity is chosen as the network resource considered in admission control, and it is specified by the proposed spare capacity curve (section 3.2.1). By taking advantage of the spare capacity curve, our CAC algorithm only needs to know

the new connection's attributes (e.g., traffic parameters and QoS requirements) and the current spare capacity curve in order to make an accept / reject decision.

The design of our CAC algorithm requires the following information (Fig. 7-1):

1) The traffic parameters of the new connection j. The new traffic flow is modeled by the MLB traffic model with parameters $(\sigma_{j,n}, \rho_{j,n})$ for $n = 1, 2, \ldots, N_j$ (section 2.2.1).

2) The QoS requirements of the new connection j: the cell loss ratio of zero and the end-to-end delay bound d_j (section 3.1).

3) The available link capacity, which is described by the spare capacity curve with parameters (u_h, t_h) for $h = 1, 2, \ldots, H$ (section 3.2.1).

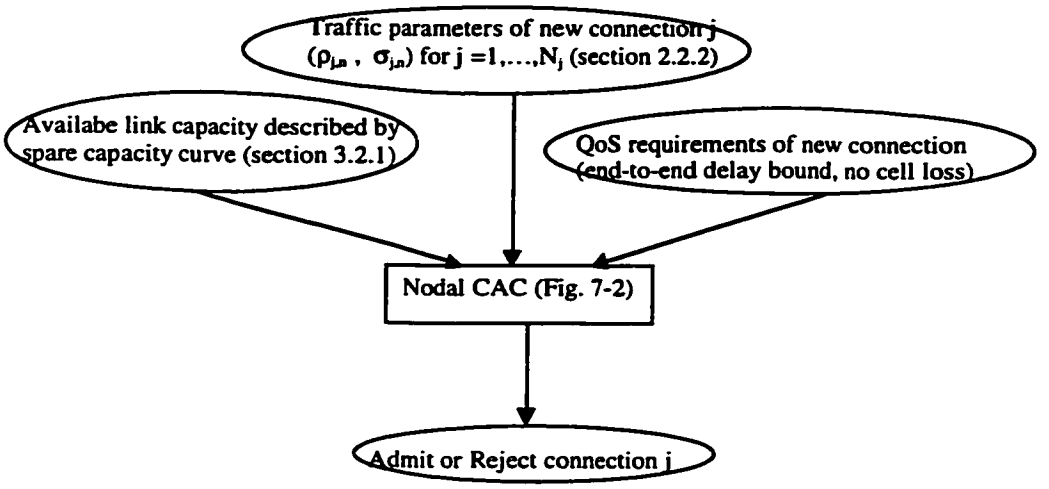


Fig. 7-1 Input and Output Parameters of CAC Algorithm

The CAC algorithm has been presented in section 3.3. Here, CAC at the node m is flow-charted in Fig. 7-2 (a) and (b). When a Request message of the new connection j is received by node m, the procedure in Fig. 7-2 (a) will be invoked. When an Accept message of the new connection j is received, the procedure in Fig. 7-2 (b) will be invoked.

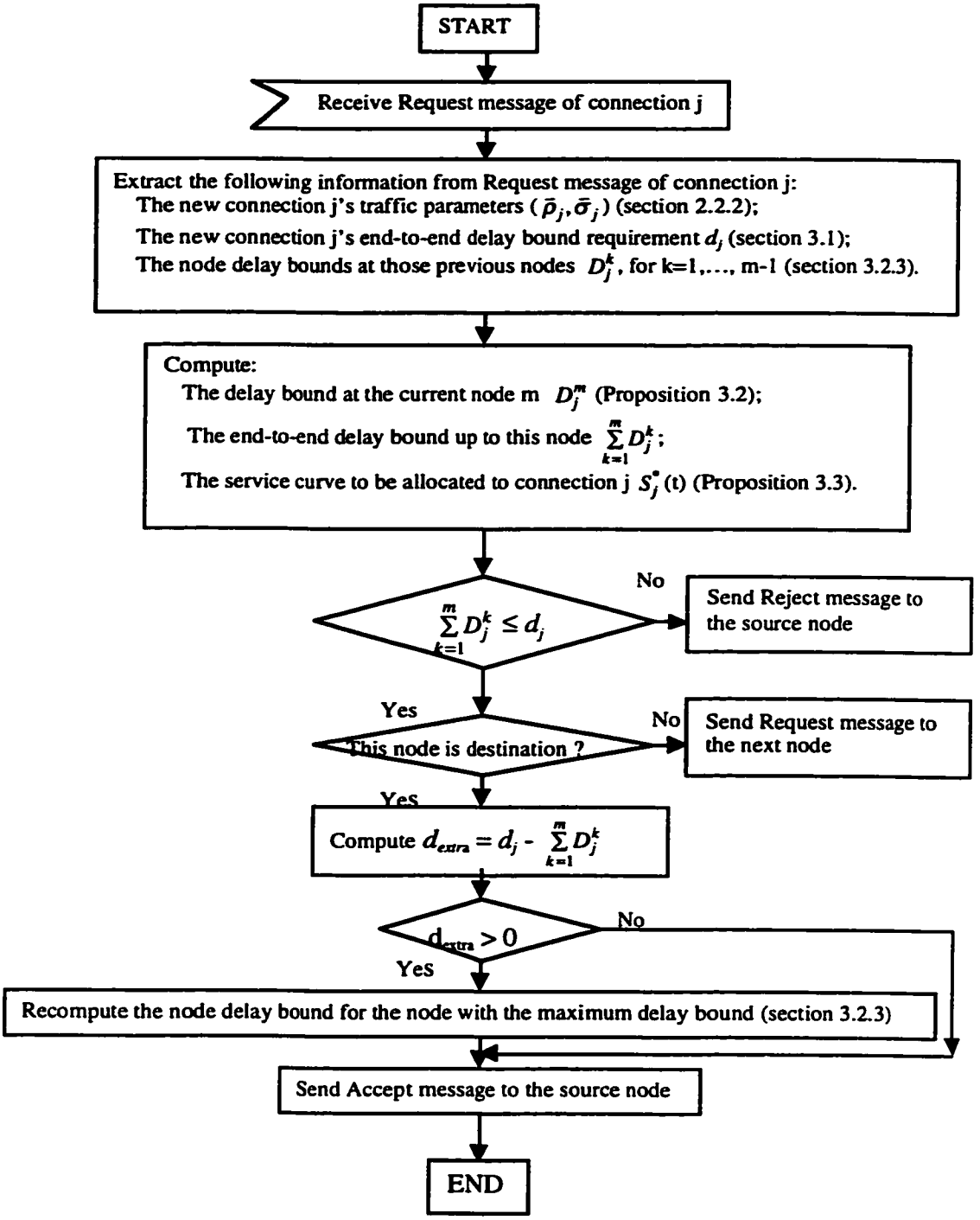


Fig. 7-2 (a) Admission Control Procedure invoked by the Request Message

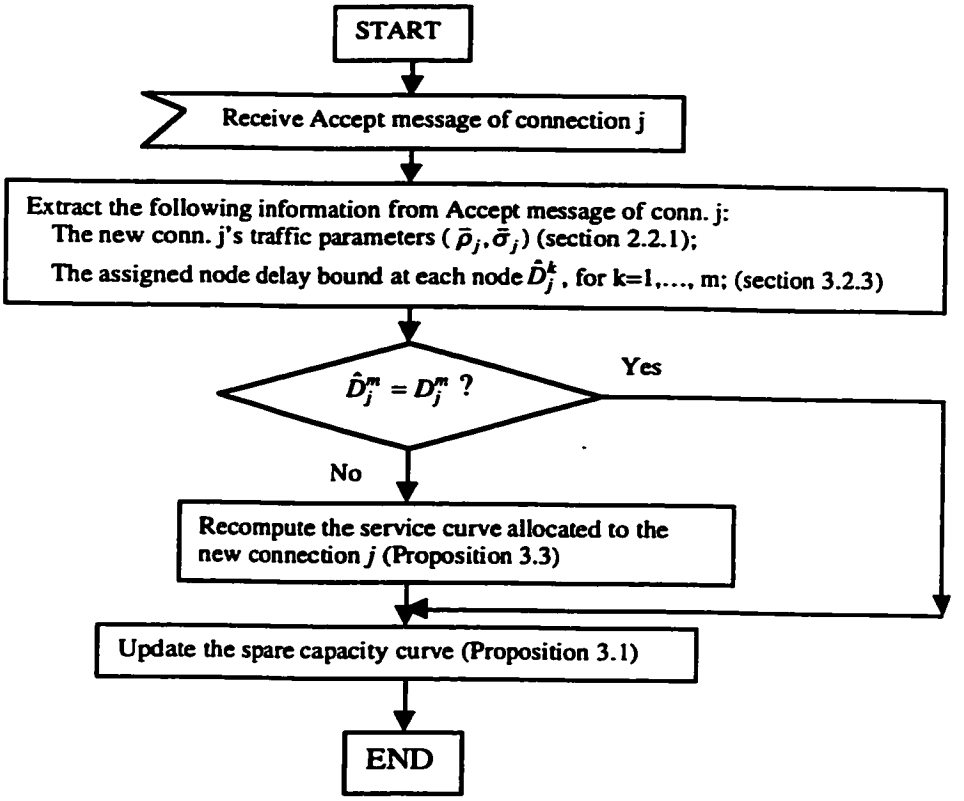


Fig. 7-2 (b) Admission Control Procedure invoked by the Accept Message

7.3 Design Guideline for Lossless Traffic Shaper

The traffic shapers designed in Chapter 4 can be used at the network entrance or inside the network. They are MLB / DLB shapers (section 2.3) that consist of multiple (two for DLB) leaky buckets in series. These shapers support the deterministic service (Chapter 4) with zero cell loss guaranteed.

The design of the lossless shapers requires the following information (Fig. 7-3):

1) The traffic parameters of the new connection j. The new traffic flow is modeled by the MLB traffic model (section 2.2.1) with parameters $(\tilde{\sigma}_{j,n}, \tilde{\rho}_{j,n})$ for $n = 1, 2, \ldots, \tilde{N}_j$.

2) The requirement for the shaping delay bound D_{sj} or for the worst-case shaping buffer occupancy Q_{sj}.

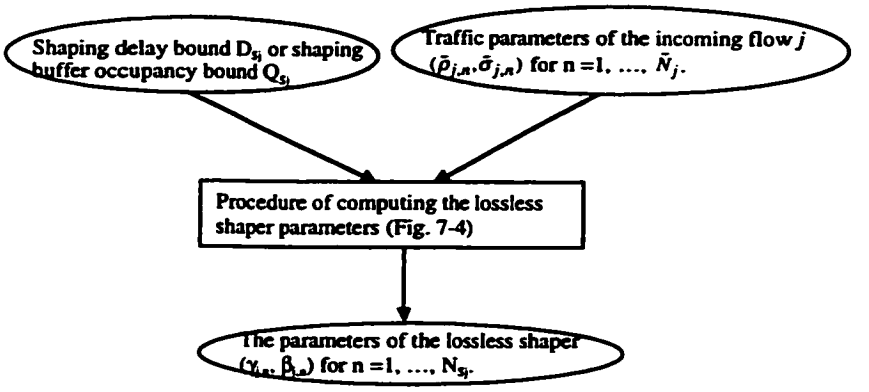


Fig. 7-3 Input and Output Parameters of the Lossless Shaper Design

The procedures of determining the parameters of the two types of lossless shaper are flow-charted in the following Fig. 7-4 (a) and (b).

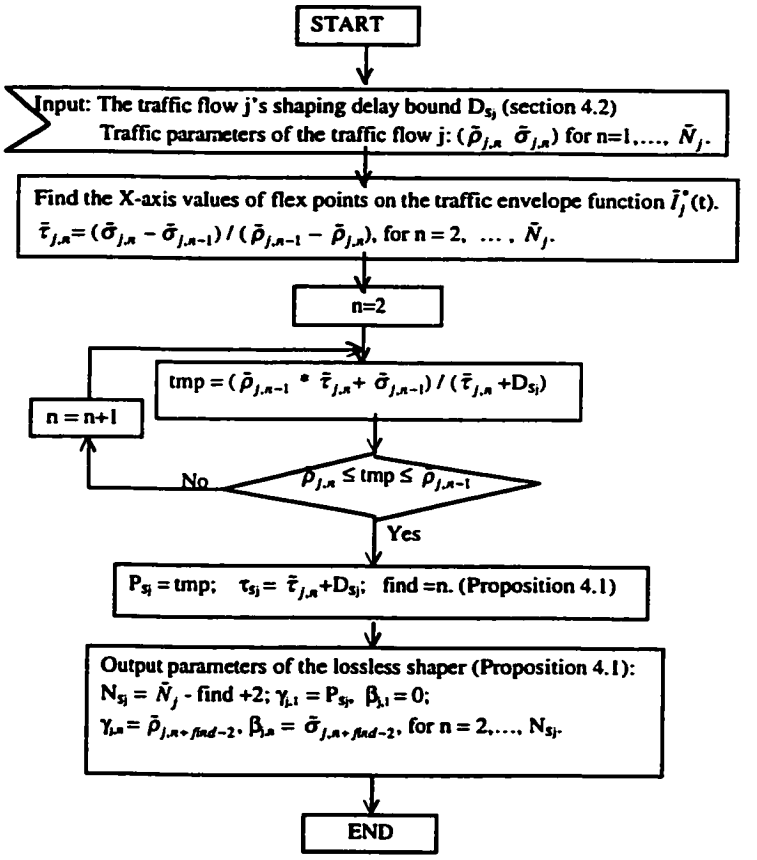


Fig. 7-4 (a) Procedure of Finding the Minimum MLB Shaper Guaranteeing the Shaping Delay Bound D_{Sj}

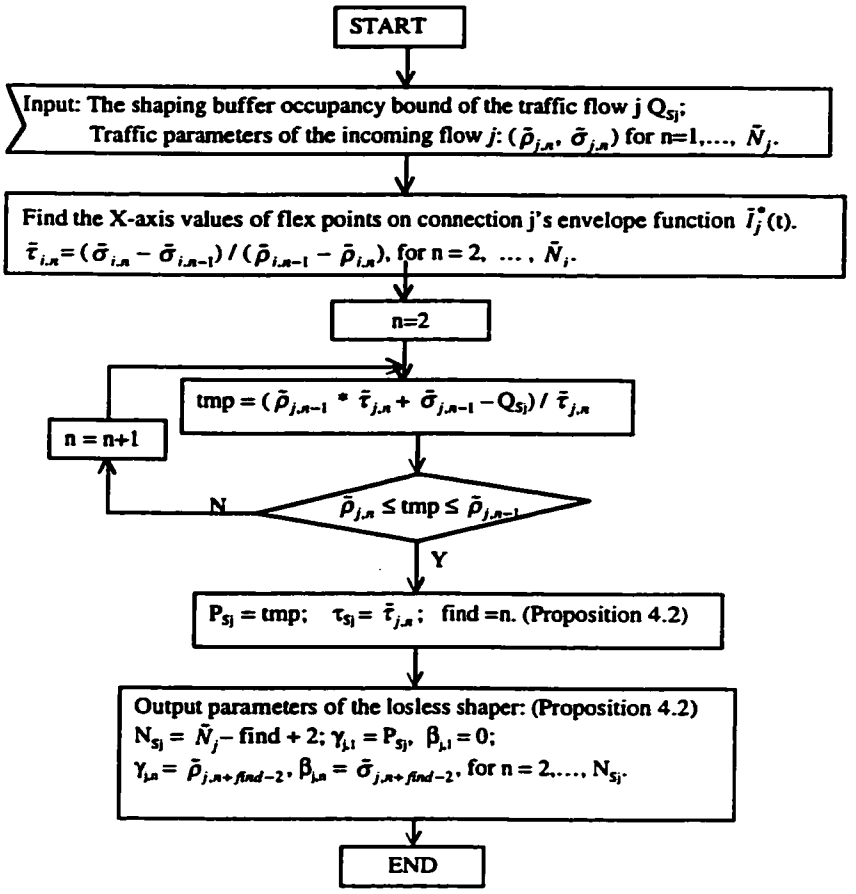


Fig. 7-4 (b) Procedure of Finding the Minimum MLB Shaper Guaranteeing the Shaping Buffer Occupancy Bound Q_{Sj}

7.4 Design Guideline for Statistical Traffic Shaper

Statistical shaping is proposed to achieve the statistical multiplexing gain at the shaper as well as at the traffic multiplexer. Different from designing the lossless shaper in section 7.3 where we searched for the parameters of the shaper envelope function, designing the statistical shaping scheme is to find a proper shaping buffer size, with which the resource utilization of statistical shaping is higher than that of lossless shaping.

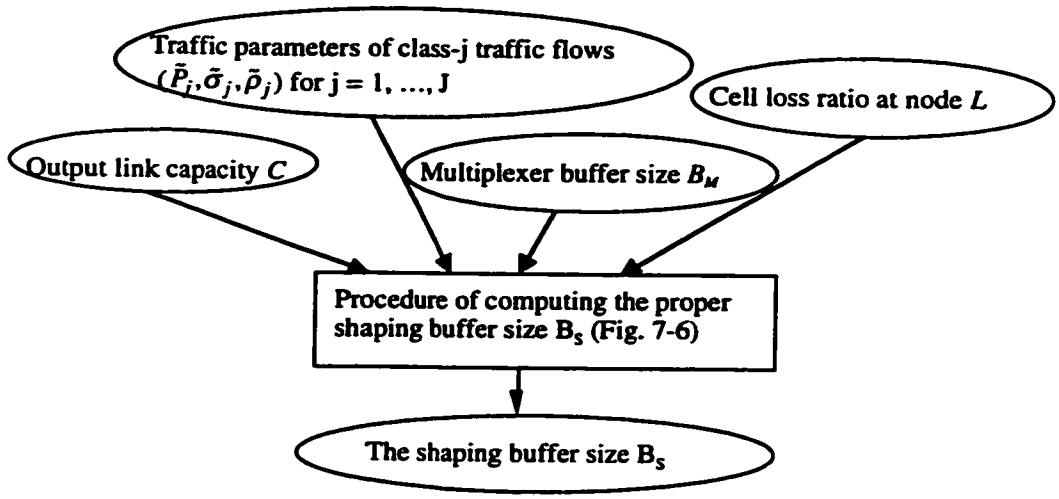


Fig. 7-5 Input and Output Parameters of Statistical Shaper Design

The design of the statistical traffic shaper requires the following information (Fig. 7-5):

1) The DLB traffic parameters of the class j traffic flows $(\tilde{P}_j, \tilde{\sigma}_j, \tilde{\rho}_j)$ (section 2.2.2);

2) The number of traffic classes J, and the number of class-j traffic flows K_j for $j = 1, \ldots, J$;

3) The QoS requirement for the cell loss ratio at the node L (section 5.3);

4) The parameters of the DLB shaper (P_S, β, γ) (section 2.3);

5) The output link capacity C (section 2.4);

6) The buffer capacity at the multiplexer B_M (section 2.4).

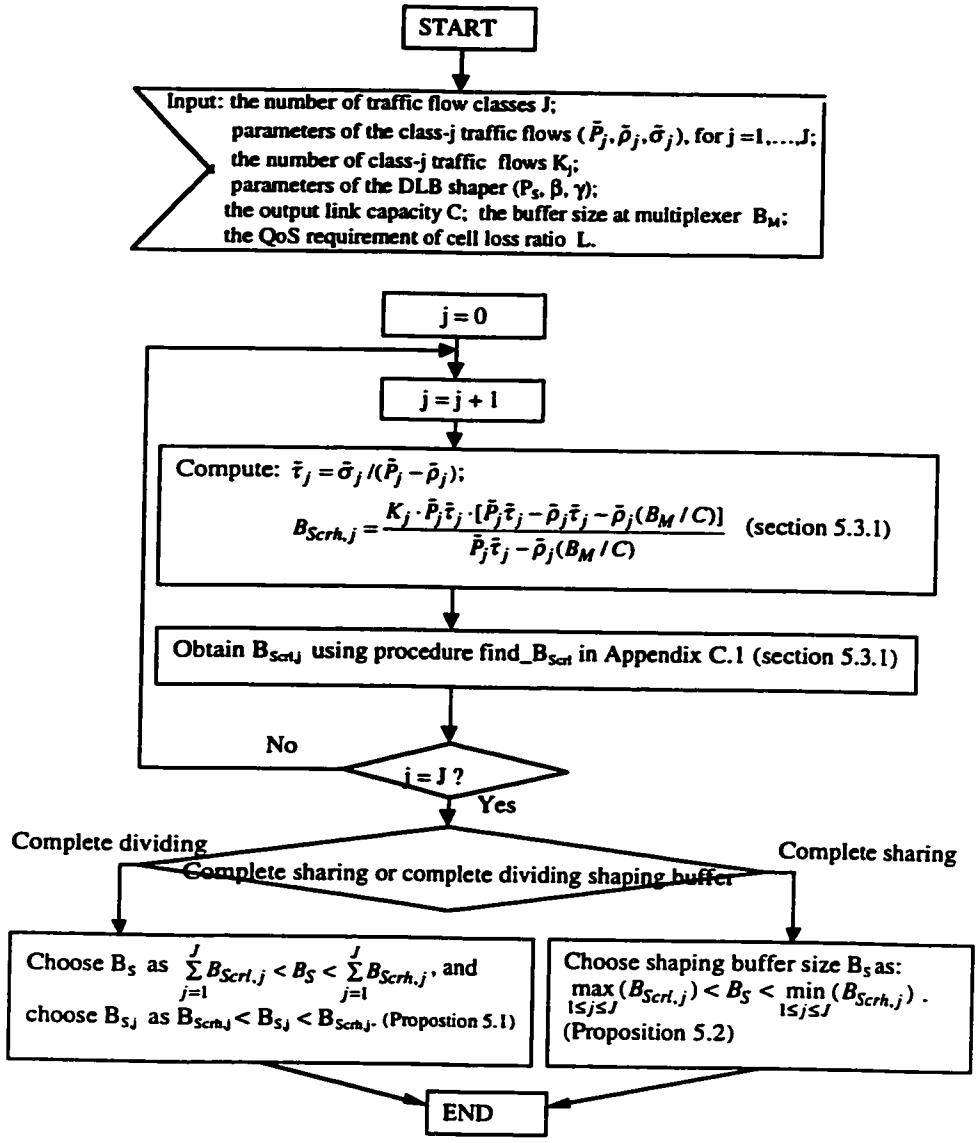


Fig. 7-6 Procedure of Finding the Proper Shaping Buffer Size

The above Fig. 7-6 is the flow chart for choosing the proper shaping buffer size. It is based on the analysis in sections 5.2 and 5.3.

7.5 Guideline for Obtaining End-to-End Performance

Through both analysis and simulation, we obtained in Chapter 6 the end-to-end performance of an individual VBR connection. Based on the work there, we provide the guideline on how to obtain the end-to-end performance for a VBR traffic flow.

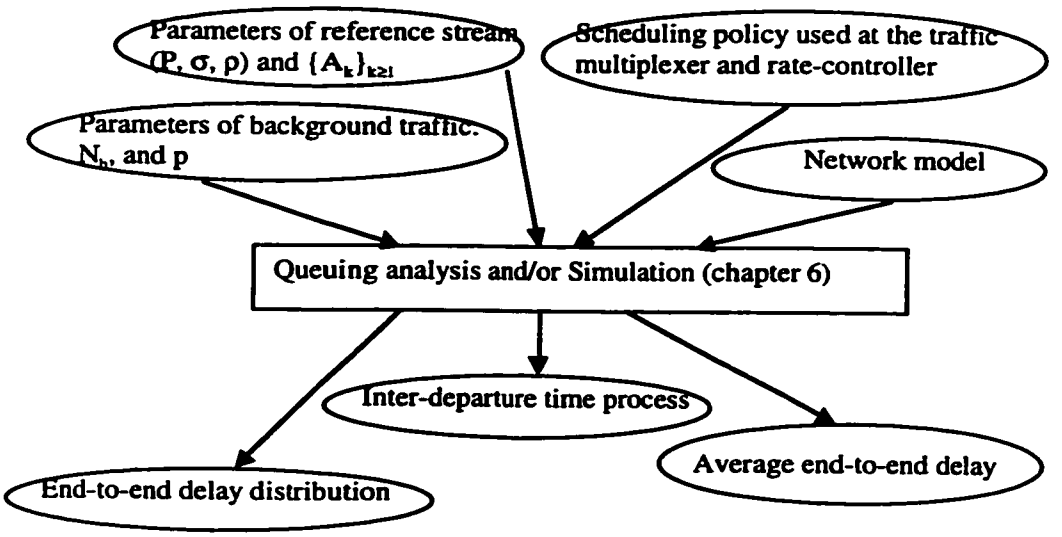


Fig. 7-7 Input and Output Parameters of End-to-End Performance Study

Obtaining the end-to-end performance requires the following information (Fig. 7-7):

1) The network model. In Chapter 6, the end-to-end QoS measures were obtained for both the RCS network and the conventional network (sections 6. 4 and 6.5).

2) Traffic scheduling policy used at each network component. In Chapter 6, the proposed leaky bucket mechanism is used at every rate-controller, and the FCFS policy is used at every multiplexer.

3) Traffic parameters of the reference stream. The reference stream is specified by both the DLB model (P, σ, ρ) (section 2.2.2) and the inter-arrival time process $\{A_k\}_{k\geq 1}$ (section 2.2.3).

4) The parameters of the background traffic. The background traffic is modeled by a Bernoulli arrival process, where both the number of background streams N_b and the traffic load of each background stream p are used to specify the characteristics (section 2.2.4).

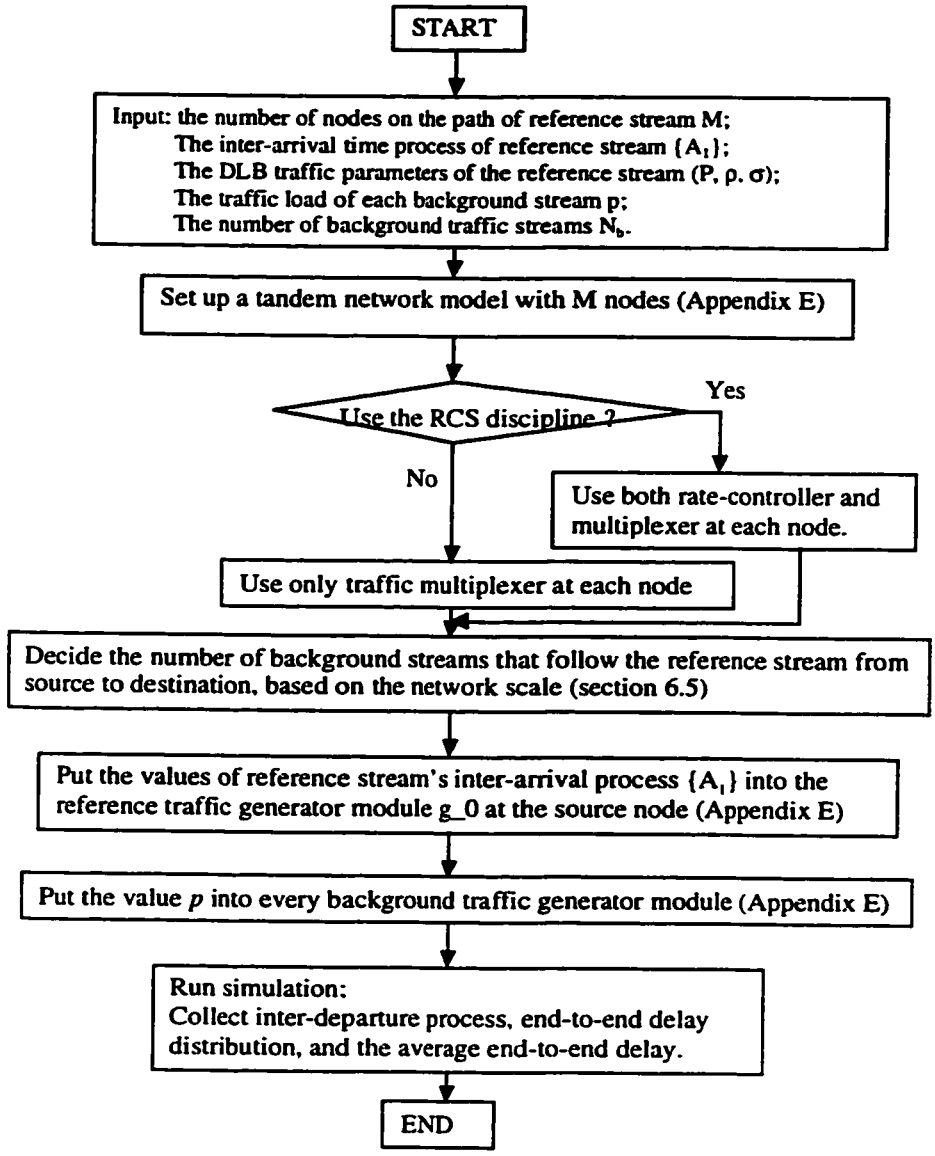


Fig. 7-8 The Procedure to Obtain End-to-End Performance Through Simulation

Fig. 7-8 gives the procedure of obtaining the end-to-end performance through simulation. The steps of obtaining the end-to-end performance through analysis have been presented in section 6.4.

From the performance evaluation in chapter 6, we obtain the following general observations that a network designer should be aware of when provisioning end-to-end VBR connections.

1) Traffic multiplexing may incur traffic pattern distortion to the traffic streams. When no traffic shaping is applied inside the network, the traffic pattern becomes more distorted as the stream traversing more nodes.

2) When the traffic shaping technique is used at each node, it helps to keep the traffic characteristics and alleviate the traffic distortion incurred by multiplexing. Therefore, network designers should consider applying traffic shaping technique at the nodes inside the network in order to prevent congestion and support satisfying network performances.

3) When traffic shaping is applied at each node along the path, the end-to-end delay of an individual connection is longer than that when no traffic shaper is used. Therefore, the network designers should keep in mind that there is trade-off to use traffic shaping. It helps to keep the traffic characteristics in terms of the inter-cell time distribution (Fig. 6-9), but it usually induces larger end-to-end delay (Fig. 6-11).

7.6 Chapter Summary and Remarks

In this chapter, we provided the guidelines on controlling VBR traffic based on the research results in the previous chapters. These design guidelines showed the network designers how to use the VBR traffic control schemes, and how to obtain the end-to-end performance.

Chapter 8

Conclusion

Research has been conducted in this thesis to pursue the objectives of VBR traffic control: to prevent network congestion, to support guaranteed QoS, and to obtain high utilization of network resources. First, an admission control algorithm has been presented for the real-time VBR Traffic, in which the service and the spare capacity curve are two important features. With the proposed approaches to allocate the service curve and to update the spare capacity curve, our algorithm performs better than the previous ones in terms of both the network bandwidth utilization and the computational time needed. Second, two lossless MLB shapers have been designed that can guarantee certain requirements for the shaping buffer occupancy or delay bound. We have also analyzed the node delay performance for the minimum MLB shaper guaranteeing the shaping delay bound, and showed that the node delay bound of this minimum MLB shaper is not larger than that when no traffic shaping is used. Third, a statistical traffic shaping scheme has been put forward. We showed that statistical shaping can achieve a higher level of resource utilization than lossless shaping when we applied certain design criteria at the network node design. Fourth, the impact of rate-controlling on an individual connection's end-to-end QoS has been analyzed. A new operating mechanism was presented for the leaky-bucket rate-controller, which has the major advantage of allowing a simple embedded Markov Chain analysis. The queuing performance of the traffic multiplexer was analyzed by using the matrix analysis technique. Simulations were carried out to evaluate the end-to-end performance in both the conventional and the RCS networks. The simulation results showed that the RCS discipline helps to keep the traffic characteristics and reduce the distortion induced by multiplexing, but

this benefit comes at the expense of a slightly longer end-to-end delay. Fifth, based on the research results, we have provided the guidelines and design criteria on controlling the VBR traffic.

The proposed VBR traffic control schemes can be applied directly in the ATM networks, because they were designed with the consideration of two major features of ATM networks, i.e., the fixed size cell and the connection-oriented network. These control schemes may need some changes in order to be applied in the other networks, e.g., the Internet.

In an ATM domain, the lossless and statistical shapers can be applied at any network node. It is not necessary that a shaper is installed in every node or that the same kind of shaper must be installed at every node. But the proposed admission control algorithm has the limitation that the same version of the software implementing the algorithm must be installed in every node in order for it to work properly.

There are many future works in the area of traffic control. For example, to alleviate both the limitation of application scope and the limitation of software implementation as mentioned above. One interesting extension of the ATM traffic management is the traffic management in MPLS (Multi-Protocol Label Switching) networks [AMAO99]. For instance, the differentiated services [NiCa00], and QoS routing [GKOP99], etc.

References

[AMAO99] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, "Requirements for traffic engineering over MPLS", IETF RFC 2702, September 1999.

[ATMF96] ATM Forum Technical Committee, "Traffic Management Specification", af-tm-0056.0000, Version 4.0, April, 1996.

[Berg91] A. W. Berger, "Performance analysis of a rate-control throttle where tokens and jobs queue", *IEEE JSAC* Vol. 9, No. 2, pp. 165-170, Feb. 1991.

[Buck90] J. A. Bucklew, *Large deviation techniques in decision, simulation, and estimation*, John Wiley & Sons, 1990.

[Chan94] C.-S. Chang, "Stability, queue length, and delay of deterministic and stochastic queuing networks", IEEE Trans. on Automatic Control, vol.39, no.5, May, 1994, pp.913-931.

[Cruz91a] R. L. Cruz, "A calculus for network delay, part I: network elements in isolation", *IEEE Trans. on Information Theory*, Vol. 37, No. 1, pp. 114-131, Jan. 1991.

[Cruz91b] R. L. Cruz, "A calculus for network delay, part II: network analysis", *IEEE Trans. on Information Theory*, Vol. 37, No. 1, pp. 132-141, Jan. 1991.

[Cruz95] R. L. Cruz, "Quality of service guarantees in virtual circuit switched networks", *IEEE JSAC*, vol.13, no. 6, Aug. 1995, pp. 1048-1056.

[Daig92] J. N. Daigle, "Queuing theory for telecommunications", Addison - Wesley, 1992.

[ElMi91] A. Elwalid, D. Mitra, "Stochastic fluid model in the analysis of access regulation in high speed networks", *proc. IEEE GLOBECOM'91*, pp. 1626-1632.

[ElMi97] A. Elwalid, D. Mitra, "Traffic shaping at a network node: theory, optimum design, admission control", *Proc. IEEE INFOCOM'97*, pp. 445-455.

[ElMW95] A. Elwalid, D. Mitra, R. Wentworth, "A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node", *IEEE JSAC*, Vol. 13, No. 6, pp. 1115-1127, Aug. 1995.

[FeVe90] D. Ferrari, D. C. Verma, "A scheme for real-time channel establishment in wide-area networks", *IEEE JSAC*, vol. 8, no. 3, Apr. 1990, pp. 368-379.

[FiKT97] V. Firoiu, J. Kurose, D. Towsley, "Efficient admission control for EDF schedulers", *in Proc. IEEE INFOCOM'97*, pp. 310-317.

[GaWi94] M. Garrett, W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic", in Proc. ACM SIGCOMM'94, Aug. 1994.

[GGPS96] L. Georgiadis, R. Guerin, V. Peris, K. N. Sivarajan, "Efficient network QoS provisioning based on per node traffic shaping", *IEEE/ACM Trans. on Networking*, Vol. 4, No. 4., pp. 482-500, Aug. 1996.

[GiHu91] R. J. Gibbens, P. J. Hunt, "Effective bandwidths for the multi-type UAS channel", *Queueing Syst.*, Vol. 9, pp. 17-28. 1991.

[GKOP99] R. Guerin, S. Kamat, A. Orda, T. Przygienda, D. Williams, "QoS routing mechanisms and OSPF extensions", IETF RFC 2676, August 1999.

[Gole90] S. J. Golestani, "A Stop-and-Go queuing framework for congestion management", *proc. SIGCOMM'90*, pp. 8-18, Sept. 1990.

[Graf97] M. Graf, "VBR video over ATM: reducing network resource requirements through endsystem traffic shaping", *Proc. IEEE INFOCOM'97*, pp. 48-57.

[GrBo99] M. Grossglauser, J.-C. Bolot, "On the relevance of long-range dependence in network traffic", IEEE/ACM Transactions on Networking, Vol.7, No. 5 , Oct. 1999 , pp. 629-640.

[GuAN91] R. Guerin, H. Ahmadi, and M, Nagshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks", *IEEE JSAC*, Vol. 9, No. 7, pp. 968-981, 1991.

[HeLu86] H. Heffes, D. M. Lucantoni, "A Markov modulated characteristics of packetized voice and data traffic and related multiplexer performance", IEEE JSAC, Vol. 4, no. 6, pp. 856-868, Sept. 1986.

[Kell91] F. P. Kelly, "Effective bandwidths at multi-class queues", *Queueing Syst.*, Vol. 9, pp. 5-16, 1991.

[KeMN97] P. Key, R. Macfadyen, M. Nilsson, "VBR video, policing, and dimensioning", ATM Forum / 97 – 0308, April, 1997.

[Knig96] E. W. Knightly, "H-bind: a new approach to providing statistical performance guarantees to VBR traffic", in *Proc. IEEE INFOCOM'96*, pp. 1091-1099.

[KnZh97] E. W. Knightly, H. Zhang, "D-BIND: an accurate traffic model for providing QoS guarantees to VBR traffic", IEEE/ACM Trans. on Networking, vol.5, no. 2, April, 1997, pp. 219-231.

[Kuro92] J. Kurose, "On computing per-session performance bounds in high-speed multi-hop computer networks", *Performance Evaluation Review*, Vol. 20, No. 1, pp. 128-139, June, 1992.

[KETB92] H. Kroner, M. Eberspacher, T. H. Theimer, P. J. Kuhn, U. Briern, "Approximate analysis of the end-to-end delay in ATM networks", *proc. IEEE INFOCOM'92*, pp. 978-986.

[LaCY96] S. S. Lam, S. Chow, D. K. Y. Yau, "A lossless smoothing algorithm for compressed video", *IEEE/ACM Trans. on Networking*, Vol. 4, No. 5., pp. 697-708, Oct. 1996.

[LaLi97] Wing-cheong Lau, San-qi Li, "Statistical multiplexing and buffer sharing in multimedia high-speed networks: a frequency-domain perspective", IEEE/ACM Transactions on Networking, Vol.5, No.3, June 1997 , Page(s): 382 –396.

[LaSt97] R. Landry, I. Stavrakakis, "Study of delay jitter with and without peak rate enforcement", *IEEE/ACM Trans. on Networking*. Vol. 5, No. 4., pp. 543-553, August, 1997.

[LeLD96] Tsern-Huei Lee; Kuen-Chu Lai; Shii-Tyng Duann "Design of a real-time call admission controller for ATM networks", IEEE/ACM Transactions on Networking, Vol.4, No. 5, Oct. 1996, pp. 758 –765.

[LiTo95] Zhen Liu; Don Towsley, "Burst reduction properties of rate-control throttles: downstream queue behavior", IEEE/ACM Transactions on Networking, Vol. 3, No. 1, Feb. 1995 , pp. 82 –90.

[LiTs97] M. Li, Z. Tsai, "Design and analysis of the GCRA traffic shaper for VBR services in ATM networks", *Proc. IEEE ICC'97*, pp. 210-214.

[LiWF96] J. Liebeherr, D. E. Wrege, D. Ferrari, "Exact admission control for networks with a bounded delay service", *IEEE/ACM Trans. on Networking*, vol.4,no.6, Dec.1996, pp. 885-901.

[MiMo95] D. Mitra, J. Morrison, "Multiple time scale regulation and worst case processes for ATM network control", *Proc. of the 34th Conference on Decision and Control*, pp. 353-358, Dec. 1995.

[Neut89] M. F. Neuts, "Structured stochastic Matrices of the M/G/1 type and their applications", Marcel Dekker, New York, 1989.

[NiCa00] K. Nichols, B. Carpenter, "Definition of differentiated services behavior aggregates and rules for their specification", Internet draft of IETF (Internet Engineering Task Force), Feb. 2000.

[OhMM91] Y. Ohba, M. Murata, H. Miyahara, "Analysis of interdeparture processes for bursty traffic in ATM networks", *IEEE JSAC* Vol. 9, No. 3, 1991, pp. 468 - 476, April, 1991.

[Mil97] Mil3 Inc., "OPNET modeler", Version 2, Vol. 1 – 12, 1997.

[PaGa93] A. K. Parekh, R.G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case", *IEEE/ACM Trans. on Networking*, vol.1,no.3, June,1993, pp.344-357.

[PaGa94] A. K. Parekh, R.G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case", *IEEE/ACM Trans. on Networking*, vol.2, no.2, April, 1994, pp. 137-150.

[PeEl96] H.G. Perros, K.M. Elsayed, "Call admission control schemes: a review", IEEE Communications Magazine, Vol. 34, No. 11, Nov. 1996 , pp. 82-91.

[PFTV86] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, "Numerical Recipes", Cambridge University Press, 1986.

[RaKZ96] A. Raha, S. Kamat, W. Zhao, "Admission control for hard real-time connections in ATM LANs", in *Proc. IEEE INFOCOM'96*, San Francisco, CA, pp.180-188.

[Rose95] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems", Institute of Computer Science, University of Wurzburg, Germany, Tech. Rep. 101, Feb. 1995.

[ShYa98a] T. Shan, O. W. W. Yang, "An admission control scheme for the real-time VBR traffic in the ATM network: deterministic bandwidth allocation", *Proc. IEEE International Conference on Communications (ICC'98)*, Atlanta, GA, June, 1998, pp. 1505-1509.

[ShYa98b] T. Shan, O. W. W. Yang, "Designing an Appropriate Traffic Shaping Scheme to Reduce Network Resource Requirements", *Proc. IEEE Conference on Enterprise Networking 1998 (ENCOM'98)*, Atlanta, GA, June, 1998, pp. 2A.1-9.

[ShYa98c] T. Shan, O. W. W. Yang, "Proposals on Lossless and Statistical Traffic Shaping", *Proc. IEEE Military Communications Conference (MILCOM'98)*, Bedford, MA, Nov. 1998, pp. 772-776.

[ShYa98d] T. Shan, O. W. W. Yang, "A Rate-Controller for the ATM Network", *Proc. IEEE International Conference on Computer Communications and Networks (IC3N'98)*, Lafayette, LA, Nov. 1998, pp. 170-177.

[ShYa99a] T. Shan, O. W. W. Yang, "Improving Resource Utilization for the Rate-Controlled Traffic Flows in High Speed Networks", *Proc. IEEE International Conference on Communications (ICC'99)*, Vancouver, Canada, June, 1999, pp. 864-868.

[ShYa99b] T. Shan, O. W. W. Yang, "An efficient admission control scheme for the real-time VBR traffic in the ATM network: deterministic bandwidth allocation", *Computer Communications*, Elsevier Science, pp. 966-979, June, 1999.

[ShYa00] T. Shan, O. W. W. Yang, "A Rate-Controller for High Speed Networks", to appear on the journal *Telecommunication Systems*.

[SLCG93] M. Sidi, W.-Z. Liu, I. Cidon, I. Gopal, "Congestion control through input rate regulation", IEEE Trans. on Comm. Vol. 41, No. 3, pp. 471- 477, March, 1993.

[Stal94] W. Stallings, "Data and Computer Communications", Fourth Ed., Macmillan Publishing Company, 1994.

[StVa97] D. Stiliadis, A. Varma, "A general methodology for designing efficient traffic scheduling and shaping algorithms", *Proc. IEEE INFOCOM'97*, pp. 326-335.

[VeKW95] G. de Veciana, G. Kesidis, J. Walrand, "Resource management in wide-area ATM networks using effective bandwidths", IEEE JSAC, Vol. 13, No. 6 , Aug. 1995 , pp. 1081-1090.

[VeZF91] D. Verma, H. Zhang, D. Ferrari, "Guaranteeing delay jitter bounds in packet switching networks", *proc. TRICOMM'91*, pp. 35-46, Apr. 1991.

[WTSW97] W. Willinger, M.S. Taqqu, R. Sherman, D.V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level", IEEE/ACM Transactions on Networking, Vol.5, No.1 , Feb. 1997 , pp. 71 –86.

[WuJC97] C.-S. Wu, J.-C. Jiau, K.-J. Chen, "Characterizing traffic behavior and providing end-to-end service guarantees within ATM networks", *Proc. IEEE INFOCOM'97*, pp. 336-344.

[WKZL96] D.E. Wrege, E. W. Knightly, H. Zhang, J. Liebeherr, "Deterministic delay bounds for VBR video in packet-switching networks: fundamental limits and practical trade-offs", *IEEE/ACM Trans. on Networking*, vol.4, no.3, June, 1996, pp.352-362.

[ZhFe93] H. Zhang, D. Ferrari, "Rate-controlled static-priority queuing", *proc. IEEE INFOCOM'93*, pp. 227 - 236.

[ZhFe94] H. Zhang, D. Ferrari, "Rate-controlled service disciplines", *Journal of High Speed Networks* Vol. 3, No. 4, pp. 389-412, 1994.

[ZhKn94] H. Zhang, E. W. Knightly, "Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models", *proc. ACM SIGMETRICS'94*, pp. 211-220, May, 1994.

[ZhSh94] Q. Zheng, K. G. Shin, "On the ability of establishing real-time channels in point-to-point packet switched networks", *IEEE Trans. on Commun*. Vol.42, No. 2/3/4, Feb./March/April, 1994, pp.1096-1105.

Appendix A

Proofs of Propositions in Chapter 3

A.1 Proof of Proposition 3.1

We want to show that the spare capacity curve after the connection j is admitted should be updated as given in Proposition 3.1.

Let SC[s, s+t] and $SC'[s,s+t]$ represent respectively the number of cells that can be served by the spare capacity during a time interval of length t before and after connection j is admitted. From the definitions of $S_j^*(t)$ and $SC_*(t)$, we have

$$SC'[s,s+t] = \text{SC[s, s+t]} - S_j[s,s+t]$$

$$\geq \text{SC[s, s+t]} - S_j^*(t)$$

$$\geq SC_*(t) - S_j^*(t).$$

Thus, the updated spare capacity curve after the connection j is admitted, can be given by $SC'_*(t) = SC_*(t) - S_j^*(t)$. $S_j^*(t)$ is allocated using the method in Proposition 3. Therefore, it is always true that $S_j^*(t) \leq SC_*(t)$ for any $t \geq 0$.

A.2 Proof of Proposition 3.2

We want to show that the minimum upper bound of the scheduling delay can be computed as in Proposition 3.2.

This proof is similar to that of Theorem 2.4 in [Chan94]. Unlike [Chan94] in which the service rate is constant, the service rate in this proposition is variable.

From eqn. (3-2) and the assumption that $SC_*(t)$ is convex during [0, T_B], we have $\lim_{t\to\infty}(SC_*(t)/t) = \sup_{t\geq 1}(SC_*(t)/t) = u_H$. It means that at most u_H packets can be served by the current spare capacity per time slot.

From the definition of I_j^*, we have $\lim_{t\to\infty}(I_j^*(t)/t) = \inf_{t\geq 1}(I_j^*(t)/t) = \rho_{j,N_j}$. Thus, there are at least ρ_{j,N_j} arrivals per time slot.

Consider a discrete-time queue at the output link, and let q(t) be the number of the packets in the queue at time t. Let i(t) and sc(t) denote the number of arrivals and the number of departures during [t-1,t] respectively, where t ≥1 is arbitrarily chosen. Then,

$$q(t) = (q(t-1) + i(t) - sc(t))^+ \quad , \qquad \text{(A2.1)}$$

where $(x)^+ = \max(0, x)$.

Assume the queue is empty at time 0, and use equation (A2.1) recursively, we get

$$q(t) = \max\{0,\ i(1)-sc(1),\ i(2)+i(1)-sc(2)-sc(1),\ldots,$$

$$i(t)+i(t-1)+\ldots+i(1)-sc(t)-sc(t-1)-\ldots-sc(1)\}$$

$$= \max\{0,\ I[0,1]-SC[0,1],\ I[0,2]-SC[0,2],\ \ldots,\ I[0,t]-SC[0,t]\} \quad .$$

The second equation comes from the following definitions of I[s, s+t] and SC[s, s+t],

$$I[s,s+t] = i(s+1) + i(s+2) + \ldots + i(s+t)$$

$$SC[s,s+t] = sc(s+1) + sc(s+2) + \ldots + sc(s+t)$$

Arbitrarily choosing $s \in [1,t]$, we have

$$q(t) \geq I[0,s] - SC[0,s]$$

$$\geq I[0,s] - u_H \cdot s$$

$$\geq (\rho_{j,N_j} - u_H) \cdot s \quad .$$

(1) Since $u_H < \rho_{j,N_j}$, and both t and s are arbitrary, we have $\lim_{t \to -\infty} q(t) = \infty$. So, d_j^m can not be upper-bounded by any constant $D_j^m < \infty$.

(2) Since $u_H \geq \rho_{j,N_j}$, $\lim_{t \to -\infty} q(t) = -\infty$ or $\lim_{t \to -\infty} q(t) = 0$, thus d_j^m is finite. The proof is trivial if D_j^m=0, so we assume D_j^m>0. Assume that connection j begins its session at time 0. From the definition of D_j^m, we have

$$I_j[0,t] > SC[0, t + D_j^m - 1]$$

$$\geq SC_{\bullet}(t + D_j^m - 1) \quad .$$

From the definition of $I_j^{\bullet}(t)$, we have $I_j^{\bullet}(t) > SC_{\bullet}(t + D_j^m - 1)$. Therefore,

$$D_j^m \leq min\{\delta t: \delta t \geq 0 \text{ and } I_j^{\bullet}(t) \leq SC_{\bullet}(t + \delta t)\}$$

$$\leq \max_{t: t \geq 0} \min\{\delta t: \delta t \geq 0 \text{ and } I_j^{\bullet}(t) \leq SC_{\bullet}(t + \delta t)\}.$$

We can see that D_j^m is actually the maximum horizontal distance between $I_j^{\bullet}(t)$ and $SC_{\bullet}(t)$.

A.3 Proof of Proposition 3.3

We show that the scheduling delay bound D_j^m can be guaranteed by using the rules in Proposition 3.3 to allocate the service curve.

We first prove that $D_j^m=0$ can be guaranteed when $S_j^*(t)$ is allocated using Rule (1). From the definition of $I_j^*(t)$, we have $I_j[s,s+t] \le I_j^*(t)$, for any s, t≥0. From the definition of $S_j^*(t)$, we have $S_j^*(t) \le S_j[s,s+t]$. From Rule (1), we have $I_j[s,s+t] \le S_j[s,s+t]$, and that means during any time interval of length t, the number of arrivals from connection j is less than the number of departures from connection j. So no delay will be incurred.

Next, we prove that $0<D_j^m<\infty$ can be guaranteed when $S_j^*(t)$ is allocated using Rule (2). In order to support D_j^m, the minimum service needed is $I_j^*(t-D_j^m)$. From equation (3-3), we can see that D_j^m is actually the maximum horizontal distance between $I_j^*(t)$ and $SC_*(t)$, so $I_j^*(t-D_j^m) \le SC_*(t)$. Thus, Rule (2) allocates more service capacity than that is necessary to support D_j^m for $0 < t < t_d$.

We shall prove by the method of induction that the updated spare capacity curve can always remain to be convex. Assume there is no traffic passing through an output link at node m before time 0, then the spare capacity curve function at time 0 is $SC_*(t) = t$. Without loss of generality, we assume that the first connection Request arrives at time 0. Then there are three possible cases as illustrated in Fig. A-1.

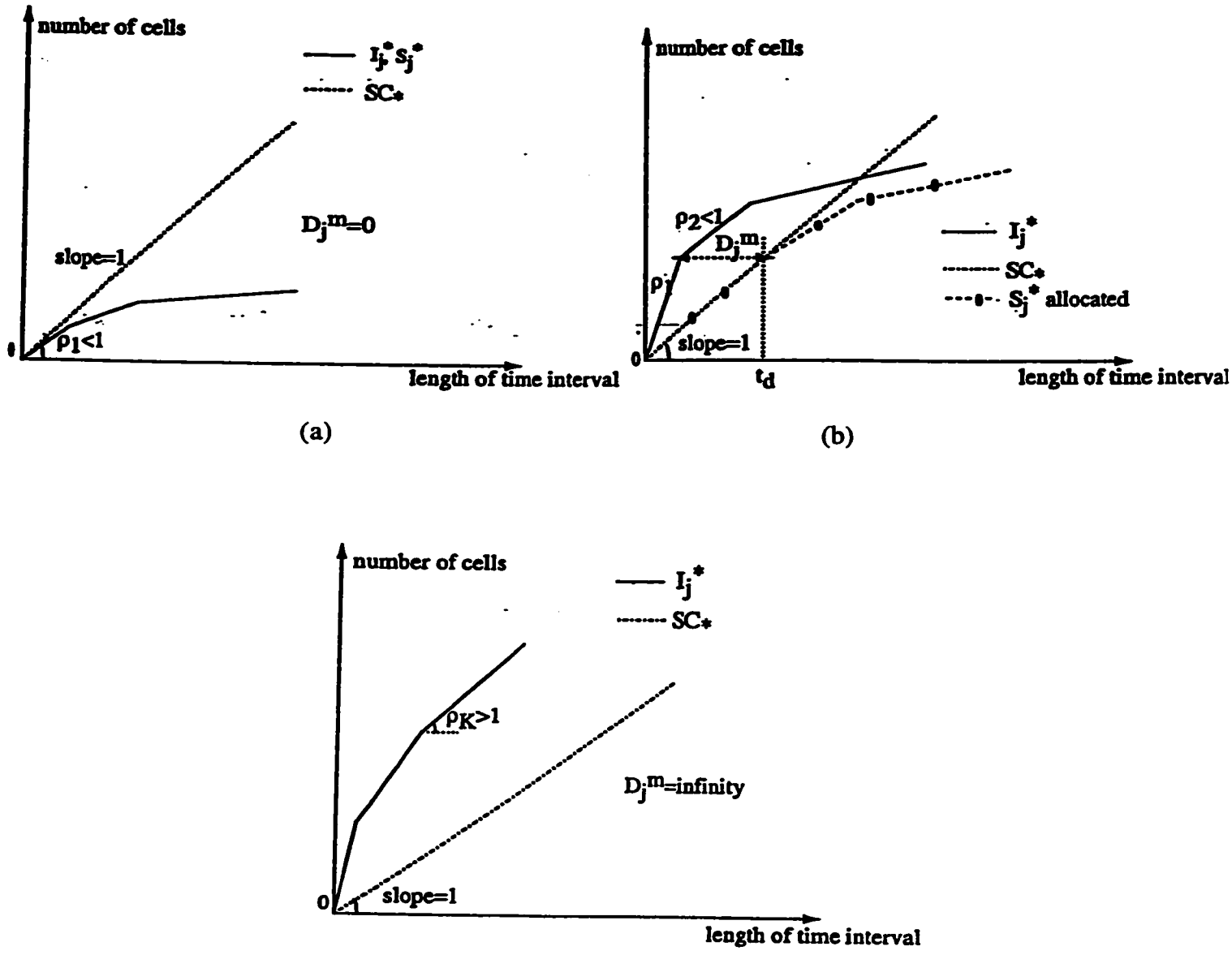


Fig. A-1 Various Cases of Bandwidth Allocation, at the Beginning of Busy Period

Case (a): $1 \geq \rho_{1,1}$

As shown in Fig. A-1 (a), no node delay will be incurred on the new stream due to the limitation of the link capacity, i.e. $D_j^m=0$. Then the service curve can be allocated according to the Rule (1).

After admitting the first connection whose traffic constraint function is described by $\{(\rho_{1,n}, \tau_{1,n}) | n = 1, ..., N_1\}$, the slope of the *n*th line segment in the spare capacity curve will be updated to $1 - \rho_{1,n}$, for $\tau_{1,n} \le t < \tau_{1,n+1}$. Because the traffic constraint function of the first connection is a concave function, i.e. $\rho_{1,1} \ge \rho_{1,2} \ge ... \ge \rho_{1,N_1}$, so $1 - \rho_{1,1} \le 1 - \rho_{1,2} \le ... \le 1 - \rho_{1,N_1}$. That is, $SC'_*(t)$ must remain piecewise linear and convex.

Case (b): $1 < \rho_{1,1}$ and $\rho_{1,n-1} > 1 \ge \rho_{1,n}$, where $2 \le n \le N_1$

As shown in Fig. A-1 (b), the scheduling delay incurred on the connection is larger than 0, but can be bounded by $D_1^m < \infty$. The service curve can be allocated according to the Rule (2). Using Proposition 3.1, we have

$$SC'_*(t) = 0 \qquad \text{for } 0 < t \le t_d;$$

$$SC'_*(t) = (1 - \rho_{1,n}) \cdot (t - t_d) \qquad \text{for } t_d \le t < \tau_{1,n+1} + D_1^m;$$

$$SC'_*(t) = (1 - \rho_{1,i})t + \sum_{l=n+1}^{i} (\rho_{1,l} - \rho_{1,l-1})\tau_{1,l} + (\rho_{1,i} - \rho_{1,n})D_1^m - (1 - \rho_{1,n})t_d,$$

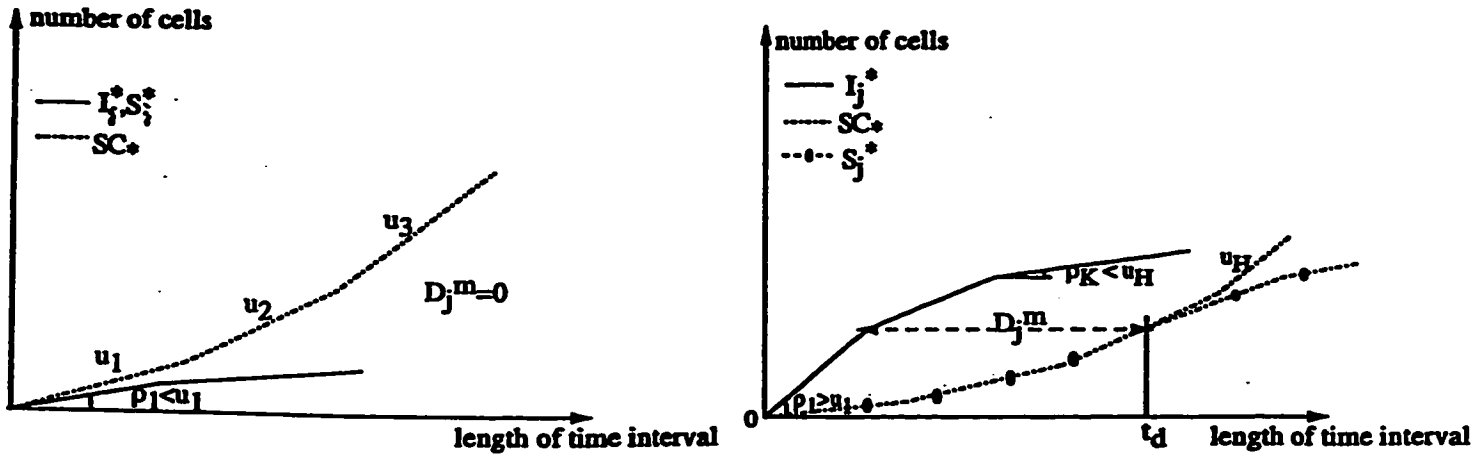
$$\text{for } \tau_{1,i} + D_1^m \le t < \tau_{1,i+1} + D_1^m,\ i = n+1, ..., N_1.$$

Since $0 < 1 - \rho_{1,n} \le 1 - \rho_{1,n+1} \le ... \le 1 - \rho_{1,N_1}$, $SC'_*(t)$ remains piecewise linear and convex.

Case (c): $\rho_{1,N_1} > 1$

As shown in Fig. A-1 (c), the node delay can not be bounded by any constant $D_j^m < \infty$. So, the connection request will be rejected, and the spare capacity curve need not be updated.

Therefore, the spare capacity curve remains piecewise linear and convex after the first connection is admitted.



(a) (b)

Fig. A-2 Various Cases of Bandwidth Allocation, When *k* Connections Exist

Next, we shall prove that the spare capacity curve remains piecewise linear and convex after the $(k+1)$st connection is admitted, given that the spare capacity curve is piecewise linear and convex after k connections have been admitted at node m. Assume that the traffic constraint function of the (k+1)st connection is described by $\{(\rho_{k+1,n}, \tau_{k+1,n}) | \, n = 1, ..., N_{k+1}\}$. Let the updated spare capacity curve SC'_* be described by $\{(u'_h, t'_h), h = 1, ..., H'\}$. There are two possible cases as shown in Fig. A-2.

Case (a): $\rho_{k+1,1} < u_1$

As shown in Fig. A-2 (a), $D^m_{k+1}=0$ and S^*_{k+1} is allocated using Rule (1). The spare capacity curve is updated using Proposition 3.1, so we have

$$SC'_*(t) = (u_h - \rho_{k+1,n})t - u_h t_h + \rho_{k+1,n}\tau_{k+1,n} + \sum_{i=1}^{h-1} u_i t_i - \sum_{i=1}^{n-1} \rho_{k+1,i}\tau_{k+1,i}$$

$$\text{for } \tau_{k+1,n} \leq t \leq \tau_{k+1,n+1} \text{ and } t_h \leq t \leq t_{h+1} \qquad \text{(A3.1)}$$

Since $u_1 < u_2 < \ldots < u_H$ and $\rho_{k+1,1} > \rho_{k+1,2} > \ldots > \rho_{k+1,N_{k+1}}$, we have $u_h - \rho_{k+1,n} \le u_h - \rho_{k+1,n+1} \le u_{h+1} - \rho_{k+1,n+1}$ and $u_h - \rho_{k+1,n} \le u_{h-1} - \rho_{k+1,n}$, thus, $SC'_*(t)$ keeps being piecewise linear and convex.

Case (b): $\rho_{k+1,1} > u_1$ and $\rho_{k+1,N_{k+1}} < u_H$

As shown in Fig. A-2 (b), $0 < D_j^m < \infty$ and S_j^* is allocated using Rule (2). $SC'_*(t)=0$, for $t \le t_d$; $SC'_*(t)$ can be obtained using equation (A3.1). In the same way as above, we can verify that the updated spare capacity curve keeps being piecewise linear and convex.

A.4 Proof of Proposition 3.4

We show that we can use the convex lower bound of the spare capacity curve to compute the scheduling delay bound in admission control.

Using Proposition 3.1, d_j^m is upper-bounded by

$$D_j^m = \max_{t:t\ge 0} \min\{\delta t:\ \delta t \ge 0 \text{ and } I_j^*(t) \le SC_*(t+\delta t)\}$$

$$\le \max_{t:t\ge 0} \min\{\delta t:\ \delta t \ge 0 \text{ and } I_j^*(t) \le SC_*^v(t+\delta t)\}$$

$$= d^v$$

Therefore, d_j^m is upper-bounded by d^v, where, d^v is the minimum upper-bound of the scheduling delay computed using $SC_*^v(t)$.

Appendix B

Proofs of Propositions in Chapter 4

B.1 Proof of Proposition 4.1

We want to show that the shaper of Proposition 4.1 can be constructed, and that it is the minimum MLB shaper that can guarantee the shaping delay bound D_{Sj} for traffic flow j.

First, we prove that the defined τ_{Sj} always exists. This is done by observing that $y(t) = \tilde{I}_j^*(t\text{-}D_{Sj})$ can be constructed by shifting $\tilde{I}_j^*(t)$ to the right for a distance of D_{Sj}. Observed from the point (0, 0), the function $y(t) = \tilde{I}_j^*(t - D_{Sj})$ is a convex surface curved away from (0, 0), as shown in Fig. 4-1. By the basic geometry, at least one tangent line can be drawn from (0, 0) to y(t), so $(\tau_{Sj}, y(\tau_{Sj}))$ is actually the tangent point with the smallest $\tau_{Sj} > 0$ and $y(\tau_{Sj})>0$.

Secondly, we show the concavity of $A_j(t)$, which is the sufficient condition for it to be a MLB shaper function. From $A_j(t)$ definition, it is concave for $t \geq \tau_{Sj}$. From the definition of τ_{Sj} and P_{Sj}, we have that $A_j(t)$ is concave for $t \geq 0$.

Thirdly, we prove that the delay incurred on the traffic flow j when it is shaped using a shaper with the envelope function $A_j(t)$ is upper-bounded by D_{Sj}. Let the traffic shaper with the envelope function $A_j(t)$ consists of N_{Sj} $(\beta_{j,n}, \gamma_{j,n})$ - leaky buckets, then $A_j(t) = \max_{1 \leq n \leq N_{Sj}} (\gamma_{j,n} \cdot t + \beta_{j,n})$. Let $D(\tilde{I}_j^* \,||\, A_j)$ denote the shaping delay bound induced to the traffic flow j. Thus, we have

$$D(\tilde{I}_j^* \,||\, A_j) \leq \max_{1 \leq n \leq N_{Sj}} \{\frac{1}{\gamma_{j,n}} (\max_{t \geq 0} \{\tilde{I}_j^*(t) - A_j(t)\})^+\}$$

$$\leq \max_{1\leq n\leq N_{Sj}} \{\frac{1}{\gamma_{j,n}}(\max_{t\geq 0}\{\tilde{I}_j^*(t)-\gamma_{j,n}\cdot t-\beta_{j,n}\})^+\}$$

$$\leq \max_{1\leq n\leq N_{Sj}} \{(\max_{t\geq 0}\{\frac{\tilde{I}_j^*(t)-\gamma_{j,n}\cdot t-\beta_{j,n}}{\gamma_{j,n}}\})^+\}$$

$$= \max_{t\geq 0}\{(\max_{1\leq n\leq N_{Sj}}\{\frac{\tilde{I}_j^*(t)-\beta_{j,n}}{\gamma_{j,n}}\}-t)^+\}$$

$$= \max_{t\geq 0}\{(A_j^{-1}(\tilde{I}_j^*(t))-t)^+\} \quad , \qquad \text{(B-1)}$$

where

$A_j^{-1}(y)=\max_{1\leq n\leq N_{Sj}}\{\frac{y-\beta_{j,n}}{\gamma_{j,n}}\}$ is the inverse function of $A_j(t)$.

Thus, equation (B-1) indicates that the shaping delay is upper-bounded by the maximum horizontal distance between $\tilde{I}_j^*(t)$ and $A_j(t)$. From the construction of $A_j(t)$, the shaping delay is upper-bounded by D_{Sj}.

Finally, we show by contradiction that $A_j(t)$ is the minimum MLB shaper that satisfies the shaping delay bound D_{Sj}. Let $A_j^{-1}(y)$ denote the inverse function of $A_j(t)$. Assume that there is a shaper $A(t) < A_j(t)$ for $t \geq \tau_{Sj}$, such that the shaping delay bound $d_{Sj} < D_{Sj}$. Then,

$$d_{Sj}=\max_{t\geq\tau_{Sj}}\{(A^{-1}(\tilde{I}_j^*(t))-t)^+\}$$

$$\geq (t+D_{Sj})-t=D_{Sj}.$$

In another word, the contradiction exists. Therefore, $A_j(t) \leq A(t)$ for $t \geq \tau_{Sj}$. Assume that there is a shaper with envelope function $A(t) < P_{Sj}\cdot t$ for $0 \leq t < \tau_{Sj}$, such that the shaping delay bound $d_{Sj} < D_{Sj}$. Then, the y-intercept of $P_{Sj}\cdot t$ must be negative due to the concavity of A(t). This is contradicted with that A(t) is a MLB shaper. Thus, $A_j(t) \leq A(t)$ for $0 \leq t < \tau_{Sj}$.

B.2 Proof of Proposition 4.2

We want to show that the shaper of Proposition 4.2 can be constructed, and that it is the minimum MLB shaper that can guarantee the worst-case shaping buffer occupancy Q_{Sj} for traffic flow j.

First, we prove that the defined τ'_{Sj} always exists. This is done by observing that $y(t) = \tilde{I}_j^*(t) - Q_{Sj}$ can be constructed by shifting $\tilde{I}_j^*(t)$ downwards for a distance of Q_{Sj}. Observed from the point (0, 0), the function $y(t) = \tilde{I}_j^*(t) - Q_{Sj}$ is a convex surface curved away from (0, 0), as shown in Fig. 4-2. By the basic geometry, at least one tangent line can be drawn from (0, 0) to y(t), so $(\tau'_{Sj}, y(\tau'_{Sj}))$ is actually the tangent point with the smallest $\tau'_{Sj} > 0$ and $y(\tau'_{Sj}) > 0$.

Secondly, we show the concavity of $A_j(t)$, which is the sufficient condition for it to be a MLB shaping function. From the definition of $A_j(t)$, it is concave for $t \geq \tau'_{Sj}$. From the definitions of τ'_{Sj} and P'_{Sj}, we have that $A_j(t)$ is concave for $t \geq 0$.

Thirdly, we prove that the shaping buffer occupancy when the traffic flow j is shaped using a shaper with the envelope function $A_j(t)$ is upper-bounded by Q_{Sj}. Let the shaper with the envelope function $A_j(t)$ consist of N_{Sj} $(\beta_{j,n}, \gamma_{j,n})$ - leaky buckets, then $A_j(t) = \max_{1 \leq n \leq N_{Sj}} (\gamma_{j,n} \cdot t + \beta_{j,n})$.

Let $Q(\tilde{I}_j^* \,\|\, A_j)$ denote the worst-case shaping buffer occupancy of traffic flow j. Thus we have

$$
\begin{aligned}
Q(\tilde{I}_j^* \,\|\, A_j) &\leq \max_{n=1,\ldots,N_{Sj}} \{\max_{t \geq 0} (\tilde{I}_j^*(t) - \gamma_{j,n} t - \beta_{j,n})^+\} \\
&\leq \max_{t \geq 0} \{\max_{n=1,\ldots,N_{Sj}} (\tilde{I}_j^*(t) - \gamma_{j,n} t - \beta_{j,n})^+\} \\
&\leq \max_{t \geq 0} \{\tilde{I}_j^*(t) - \min_{n=1,\ldots,N_{Sj}} (\gamma_{j,n} t + \beta_{j,n}))^+\} \\
&= \max_{t \geq 0} \{(\tilde{I}_j^*(t) - A_j(t))^+\} \quad .
\end{aligned}
$$

Thus, the shaping buffer occupancy is upper-bounded by the maximum vertical distance between $\tilde{I}_j^*(t)$ and $A_j(t)$. From the construction of $A_j(t)$, the shaping buffer occupancy is upper-bounded by Q_{Sj}.

Finally, we show by contradiction that $A_j(t)$ is the minimum MLB shaper that satisfies the shaping buffer occupancy bound Q_{Sj}. Assume there is a shaper $A(t) < A_j(t)$ for $t \geq \tau'_{Sj}$, such that the shaping buffer occupancy bound $q_{Sj} < Q_{Sj}$. Then,

$$q_{Sj} = \max_{t\geq 0}\{(\tilde{I}_j^*(t)-A(t))^+\}$$

$$\geq \max_{t\geq 0}\{(\tilde{I}_j^*(t)-A_j(t))^+\} = Q_{Sj} \quad .$$

In another word, the contradiction exists. Therefore, $A_j(t) \leq A(t)$ for $t \geq \tau'_{Sj}$. Assume that there is a shaper with an envelope function $A(t) < P'_{Sj}t$, for $0 \leq t < \tau'_{Sj}$, such that the shaping buffer occupancy bound $q_{Sj} < Q_{Sj}$. Then, the y-intercept of $P'_{Sj}t$ must be negative due to the concavity of A(t), but this is contradicted with that A(t) is a MLB shaper. Thus, $A_j(t) \leq A(t)$ for $0 \leq t < \tau'_{Sj}$.

B.3 Proof of Proposition 4.3

We show that the node delay bound when using the shaper design of Proposition 4.1 is the same as that when no shaper is used.

With the assumption that the incoming traffic flow j at a shaper is characterized by the MLB traffic constraint function $\tilde{I}_j^*(t) = \min_{1\leq n\leq \tilde{N}_j}(\tilde{\rho}_{j,n}t + \tilde{\sigma}_{j,n})$, the node delay when no traffic shaper is used is upper-bounded by

$$D_j = \sup_{t\geq 0}\{\frac{\tilde{I}_j^*(t) - c_j t}{c_j}\}$$

$$= \frac{1}{c_j} \sup_{t \geq 0} \{\tilde{I}_j^*(t) - c_j t\}$$

$$= \frac{1}{c_j} \{ \tilde{I}_j^*(\tau_{j,k^\circ}) - c_j \cdot \tau_{j,k^\circ} \} \quad ,$$

where

$$\tau_{j,k^\circ} = \{ \tilde{\tau}_{j,n} \geq 0 \mid \tilde{\rho}_{j,n-1} \geq c_j \text{ and } \tilde{\rho}_{j,n} \leq c_j \} \quad .$$

When we allocate bandwidth c_j for a shaped traffic flow with peak cell rate P_{sj}, it must be true that $c_j \leq P_{sj}$. In this proposition, we use the shaper design presented in Proposition 4.1, and the shaper envelope function $A_j(t) = \min_{1 \leq n \leq N_{sj}} (\gamma_{j,n} \cdot t + \beta_{j,n})$. Let $D(A_j(t) \| c_j t)$ denote the maximum time it takes to transmit the shaped stream using the allocated bandwidth c_j, then we have

$$D(A_j(t) \| c_j t) = \sup_{t \geq 0} \{ \frac{A_j(t) - c_j t}{c_j} \}$$

$$= \frac{1}{c_j} \sup_{t \geq 0} \{A_j(t) - c_j t\}$$

$$= \frac{1}{c_j} \{A_j(\tau_{j,k^\circ}) - c_j \cdot \tau_{j,k^\circ}\} \quad ,$$

where

$$\tau_{j,k^\circ} = \{\tau_{j,n} \geq 0 \mid \gamma_{j,n-1} \geq c_j \text{ and } \gamma_{j,n} \leq c_j\}.$$

According to the construction of $A_j(t)$ in Proposition 4.1, we have $\tau_{j,k^\circ} = \tau_{j,k^\circ} + D_{sj}$ and $\tilde{I}_j^*(\tau_{j,k^\circ}) = A_j(\tau_{j,k^\circ})$. Thus, when the traffic shaper $A_j(t)$ is used, the node delay is upper-bounded by

$$D_j^* = D_{Sj} + D(A_j(t) \| c_j \cdot t)$$

$$= D_{Sj} + \frac{1}{c_j} \{A_j(\tau_{j,k^\circ}) - c_j \tau_{j,k^\circ}\}$$

$$= D_{Sj} + \frac{1}{c_j}\{\bar{I}_j^*(\tau_{j,k^\circ}) - c_j(\tau_{j,k^\circ} + D_{Sj})\}$$

$$= \frac{1}{c_j}\{\bar{I}_j^*(\tau_{j,k^\circ}) - c_j \cdot \tau_{j,k^\circ}\} = D_j \quad .$$

Appendix C

Procedures and Proofs for Chapter 5

C.1 Procedure to find B_{Scrl} the critical value design of shaping buffer size

Based on the discussions in section 5.3.1, we have formulated the following procedure *find_*B_{Srcl} to obtain B_{Scrl} the critical value of the shaping buffer size. In this procedure, two sub-procedures *find_min*P_S and *find_Kmax* are used, and they can be found after Procedure *find_*B_{Srcl}. The procedure *find_min*P_S is based on the analysis in section 5.2.2, and it finds P_S the peak rate of the shaped flow by solving the equation $\log(1 / L_S) = \Phi(\theta^*)|_{K=K_{S0},\omega_R=\gamma/P_S,b_S=(\tilde{P}-P_S)\tilde{\tau}}$, iteratively. The procedure *find_Kmax* is based on the discussions in section 5.3.1, and it obtains the maximum number of admissible connections when statistical shaping is used with $L_M = L - L_S$ and $L_S \geq 0$.

Procedure *find_*B_{Scrl}($\tilde{P}$, $\tilde{\sigma}$, $\tilde{\rho}$, K_{S0}, B_M, C)
$B_S := B_{Scrh}$;
$B_{Scrl} := B_{Scrh}$;
delta_*B_S:=10;* {*delta_*B_S affects the accuracy of B_{Scrl} and the computation time}
$\tilde{\tau} := \tilde{\sigma} / (\tilde{P} - \tilde{\rho})$;
while $((B_S > 0)$ *and* $(B_{Scrl} = B_{Scrh}))$ ***do***
    $B_S := B_S$ - *delta_*B_S;
    {find maximum number of admissible connection when L_S=0, L_M=L}
    $P_{S0} := \tilde{P} - B_S / K_{S0} / \tilde{\tau}$;
    $\beta_0 := \tilde{P} \cdot \tilde{\tau} \cdot (1 - \tilde{\rho} / P_{S0})$;
    $\gamma := \tilde{\rho}$;
    if $(P_{S0} \leq \tilde{\rho})$ ***then***
        $P_{S0} := \tilde{\rho}$; $\beta_0 := 0$;
    endif
    $e_{00} :=$ *comp_*$e_0(B_M, C, P_{S0}, \beta_0, \gamma)$; {using eqn. (2-2) }
    $L_M := L$;
    $K_0 :=$ *find_Kmax*$(L_M, e_{00}, \gamma, K_{S0}, C)$; {obtain the maximum number of admissible connections at multiplexer, when L_S=0, L_M=L.}

{find maximum number of admissible connection when L_S>0 and L_M=L-L_S}
level:=20; {the value of *level* affects both the accuracy of B_{Scrl} and computation time}
L_S:=0; $maxK_{Ls}$:=0;
while (L_S < L) **do**
L_S:= L_S+L/*level;*
P_S := *find_min*P_S(L_S, $\tilde{P}$, $\tilde{\sigma}$, $\tilde{\rho}$, K_{S0}, B_S);
β := $\tilde{P} \cdot \tilde{\tau} \cdot (1 - \tilde{\rho}/P_S)$;
e_{0Ls} := *comp_*e_0(B_M, C, P_S, β, γ); {using eqn. (2-2)}
L_M := L - L_S;
K_{Ls} := *find_Kmax*(L_M, e_{0Ls}, γ, K_{S0}, C); {obtain the maximum number of admissible connections at multiplexer, when $L_S > 0$ and $L_M = L - L_S$}
if (K_{Ls}>$maxK_{Ls}$) **then**
$maxK_{Ls}$:=K_{Ls};
endif
endwhile
if ($maxK_{Ls} \le K_0$) **then**
B_{Scrl} := B_S;
endif
endwhile
ouput (B_{Scrl}).

Procedure *find_min*P_S(L_S, $\tilde{P}$, $\tilde{\sigma}$, $\tilde{\rho}$, K_{S0}, B_S)
{This function returns the minimum value of P_S when $L_S > 0$ is allowed at shaper}
$\tilde{\tau}$:= $\tilde{\sigma}$ / ($\tilde{P}$ - $\tilde{\rho}$);
P_{S0} := P – B_S/K_{S0}/$\tilde{\tau}$;
level := 100; {the value of *level* affects the accuracy of P_S and computation time}
P_S := P_{S0} +($\tilde{P}$ - $\tilde{\rho}$) / *level;*
miny := 99999.; { assign a very large number}
Fsita := 1. + *log*(1 / L_S); {assign any number larger than *log*(1 / L_S) }

if $\left(log(1./L_S) \ge \frac{B_S}{b_S} \log(\frac{P_{S0}}{\tilde{\rho}})\right)$ **then** {It means L_S is too small to obtain statistical multiplexing gain at shaper}
P_S := P_{S0};
else
ω_R := $\tilde{\rho}$ / P_{S0};
while ((P_S > $\tilde{\rho}$) *AND* (*Fsita* > *log*(1/L_S)) **do**
P_S := P_S – (P - $\tilde{\rho}$) / *level;*
b_S := ($\tilde{P}$ - P_S). $\tilde{\tau}$;

$sita := \frac{1}{b_S}\log\frac{B_S\cdot(1-\omega_R)}{\omega_R\cdot(K_{S0}\cdot b_S - B_S)}$; {using eqn. (5-4)}

$Fsita := sita\cdot B_S - K_{S0}\cdot log(1 - \omega_R + \omega_R\cdot exp(sita\cdot b_S))$; {using eqn. (5-3)}

$y := |log(1/L_S) - Fsita|$;

if $((y \le miny)$ *AND* $(log(1/L_S) \le Fsita)$ *AND* $(P_S \ge \tilde{\rho}))$ *then*

$miny := y$;

$newP_S := P_S$;

endif

endwhile

endif

$P_S := newP_S$;

$return(P_S)$.

Procedure *find_Kmax*(L_M, e_0, γ, K_{S0}, C)

{The function returns the maximum number of admissible connections at multiplexer when $L_M = L - L_S$, $L_S \ge 0$}

$fe0 := (C / e_0)\cdot log(e_0 / \gamma)$; {compute the value of $f(e_{GLs})$ in Fig. 5-2}

if $(log(1 / L_M) \ge fe0)$ ***then*** {This means L_M is too small to obtain the statistical multiplexing gain at multiplexer}

$K_{Ls} := (int)(C / e_0)$; {compute the maximum number of admissible connections}

else

$i := (int)(C / e_0)$;

$miny := 99999.$; {assign a very large number}

$Fsita := 99999.$;

while $((i < (int)(C/\gamma))$ *AND* $(Fsita \ge log(1 / L_M))$ ***do***

i++;

$Fsita := \frac{C}{e_0}\log(\frac{C}{K_{S0}\cdot\gamma}) - (K_{S0} - \frac{C}{e_0})\log(\frac{K_{S0}(e_0-\gamma)}{K_{S0}\cdot e_0 - C})$ {using eqn.(5-5)}

$y := |\, log(1 / L_M) - Fsita\,|$;

if $((y \le miny)$ *AND* $(log(1 / L_M) \le Fsita))$ ***then***

$miny := y$;

$newi := i$;

endif

endwhile

$K_{Ls} := newi$;

endif

$return(K_{Ls})$.

C.2 Proof of Lemma 5.1

We want to show that, when lossless shaping is used, the admissible region has three cases as given in Lemma 5.1.

When $L_S = 0$, we can show from section 2.3 that the peak rate of shaped traffic stream of class j is $P_{S0,j} = \tilde{P} - b_{Sj}/\tilde{\tau}_j$. The bandwidth allocation of a class-j connection, $e_{00,j}$, can be obtained by using eqn. (2-2), with $P_{Sj} = P_{S0,j}$.

In the case of single traffic class, the features of function $F(s^*)$ were listed in section 5.3.1 as follows: (i) $\frac{\partial F(s^*)}{\partial K} = \log(\frac{Ke_0 - C}{K(e_0 - \gamma)}) < 0$; (ii) $\lim_{K \to C/e_0} \frac{\partial F(s^*)}{\partial K} = -\infty$; (iii) $\lim_{K \to C/e_0} F(s^*) = \frac{C}{e_0}\log(\frac{1}{\chi})$.

In the case of multiple traffic classes, the Chernoff's bound estimation of the loss probability at the multiplexer was obtained as follows [ElMW95]:

$$\log(L_M) \le -F_K(s^*)$$

where, $F_K(s) = sC - \sum_{j=1}^{J} K_j \log\{1 - \chi_j + \chi_j \exp(se_{0,j})\}$ (C-1)

$$F_K(s^*) = \sup_{s \ge 0} F_K(s) \quad \text{(C-2)}$$

s^* is obtained by solving the equation $\sum_{j=1}^{J} \frac{K_j \chi_j e_{0,j} \exp(se_{0,j})}{1 - \chi_j + \chi_j \exp(se_{0,j})} = C$. (C-3)

When lossless shaping is used, $L_M = L$ and $e_{0,j} = e_{00,j}$ should be put into the above three equations. Based on the operation of multiplexer in section 2.4, we know that lossless service is guaranteed at the multiplexer when $\sum_{j=1}^{J} K_j e_{00,j} = C$. Putting $\sum_{j=1}^{J} K_j e_{00,j} = C$ into equations (C-1) and (C-2), we obtained that $F_K(s^*) = \sum_{j=1}^{J} K_j \log(\frac{1}{\chi_j})$. It is also easy to show $\frac{\partial F_K(s^*)}{\partial K_j} < 0$. Thus far,

we see that various features for the case of single traffic class carry over to the case of multiple traffic classes. That is, (i) when $L_M = 0$, $\sum_{j=1}^{J} K_j e_{00,j} = C$ and $F_K(s^*) = \sum_{j=1}^{J} K_j \log(\frac{1}{\chi_j})$; (ii) when any element of vector **K** increases away from $\{\mathbf{K}: \sum_{j=1}^{J} K_j e_{00,j} = C\}$, $F_K(s^*)$ decreases.

Since there are *J* traffic classes, there are *J* corner points on the boundaries of $A(L_S=0, L_M=0)$. At the *j*th corner point, $K_i = C/e_{00,j}$ if $i = j$; $K_i=0$ otherwise. Because $\frac{\partial F_K(s^*)}{\partial K_j} < 0$ and $F_K(s^*) = \sum_{j=1}^{J} K_j \log(\frac{1}{\chi_j})$ is a linear function of **K**, $F_K(s^*)$ reaches the maximum value $\max_{1 \le j \le J}\{ \sum_{j=1}^{J} \frac{C}{e_{00,j}} \log(\frac{1}{\chi_j})\}$ at a corner point of the admissible region $A(L_S=0, L_M=0)$.

If $C < \min_{1 \le j \le J}(C_{C0,j})$ as in case (1), i.e., $\log(\frac{1}{L}) > \max_{1 \le j \le J}\{ \frac{C}{e_{00,j}} \log(\frac{1}{\chi_j})\}$, then the equation $\log(\frac{1}{L}) = F_K(s^*)$ has no solution for vector **K**. We know that $L_M = \exp(-F_K(s^*))$ should be the upper bound of the actual loss probability at the multiplexer, which is possible under this case only when the actual loss probability at the multiplexer is zero. Therefore, the boundaries of $A(L_S=0, L_M=L)$ and $A(L_S=0, L_M=0)$ coincide with each other.

If $C > \max_{1 \le j \le J}(C_{C0,j})$ as in case (2), i.e., $\log(\frac{1}{L}) < \min_{1 \le j \le J}\{ \frac{C}{e_{00,j}} \log(\frac{1}{\chi_j})\}$, then the solution to $\log(\frac{1}{L}) = F_K(s^*)$ exists for vector **K**. Since $\log(\frac{1}{L}) < \min_{1 \le j \le J}\{ \frac{C}{e_{00,j}} \log(\frac{1}{\chi_j})\}$, the element of vector **K**, $K_j(L_S=0, L_M=L) > C/e_{00,j}$ for $1 \le j \le J$. The admissible region $A(L_S=0, L_M=L)$ has *J* corner points. At the *j*th corner point, $K_i = K_j(L_S=0, L_M=L) > C/e_{00,j}$ if $i = j$; $K_i = 0$ otherwise. Thus, $A(L_S = 0, L_M=L)$ subsumes $A(L_S = 0, L_M = 0)$.

If $\min_{1\le j\le J}(C_{C0,j}) \le C \le \max_{1\le j\le J}(C_{C0,j})$ as in case (3), i.e., $\log(\frac{1}{L}) \ge \frac{C}{e_{00,j}}\log(\frac{1}{\chi_j})$ for some traffic classes, and $\log(\frac{1}{L}) < \frac{C}{e_{00,j}}\log(\frac{1}{\chi_j})$ for other traffic classes. Based on the discussion for case (1) of $C<\min_{1\le j\le J}(C_{C0,j})$, the solution of vector **K** for which $\log(\frac{1}{L}) \ge \frac{C}{e_{00,j}}\log(\frac{1}{\chi_j})$ is on the boundaries of $A(L_S = 0, L_M = 0)$. From the discussion for case (2) of $C > \max_{1\le j\le J}(C_{C0,j})$, we know that the solution of **K** for which $\log(\frac{1}{L}) < \frac{C}{e_{00,j}}\log(\frac{1}{\chi_j})$ is on the boundaries of $A(L_S = 0, L_M = L)$, which subsumes $A(L_S = 0, L_M = 0)$. Therefore, the boundaries of $A(L_S=0, L_M=L)$ and $A(L_S = 0, L_M = 0)$ coincide in part and elsewhere $A(L_S = 0, L_M = L)$ subsumes $A(L_S = 0, L_M = 0)$, which means that the boundaries of $A(L_S = 0, L_M = L)$ are non-linear.

C.3 Proof of Lemma 5.2

We wan to show that, when statistical shaping is used, the admissible region has three cases as given in Lemma 5.2.

When $L_S > 0$, the peak rate of the shaped stream of class j, $P_{SLs,j}$, can be obtained by solving the equation $\log(1/L_S) = \Phi(\theta^*)|_{K=K_{S0},\omega_R=\gamma/P_S,b_S=(\bar{P}-P_S)\bar{\tau}}$ iteratively, as stated in section 5.2.2. The bandwidth allocation of the incoming stream of class j at the multiplexer, $e_{0Ls,j}$, can then be obtained by using eqn. (2-2) with $P_{Sj} = P_{SLs,j}$.

The three cases in Lemma 5.2 give the comparison between two admissible regions when $L_M = 0$ and that when $L_M = L\text{-}L_S$, while the loss probability at the shaper is fixed as L_S. There are only two different points between this lemma and Lemma 5.1. First, the loss probability at the shaper is $L_S > 0$ in this lemma, instead of $L_S = 0$ in Lemma 5.1. Second, the bandwidth allocation

of input stream j is $e_{0Ls,j}$ in this lemma, instead of $e_{00,j}$ in Lemma 5.1. So, this lemma can be proved in the same way as Lemma 5.1, except for the above two different points.

C.4 Proof of Proposition 5.2

We show that, with different shaping buffer sizes, the relationship between the admissible region of lossless shaping and that of statistical shaping is as given in Proposition 5.2.

In Appendices C.2 and C.3, we showed that various features of $F_K(s^*)$ in the homogeneous environment carry over to the heterogeneous environment when either lossless shaping or statistical shaping is applied. Since the admissible region $A(L_S=0, L_M=L)$ has linear boundaries, as assumed in this proposition, we know from Appendix C.2 that the boundaries of $A(L_S=0, L_M=L)$ can be constructed by connecting those J corner points of $A(L_S = 0, L_M = L)$. Similarly, under the assumption that $A(L_S > 0, L_M = L - L_S)$ has linear boundaries, $A(L_S>0, L_M=L-L_S)$ can be constructed by connecting those J corner points of $A(L_S > 0, L_M = L - L_S)$.

(1) If $B_S \le \min_{1\le j\le J}(B_{Scrl,j})$, then from the definition of B_{Scrl} and Proposition 5.1, we know that $K_j(L_S > 0, L_M = L - L_S) \le K_j(L_S = 0, L_M = L)$, for $j = 1, ..., J$. From the above discussion, we obtain that $A(L_S > 0, L_M = L - L_S)$ and $A(L_S = 0, L_M = L)$ may coincide at certain corner points and elsewhere $A(L_S = 0, L_M = L)$ subsumes $A(L_S > 0, L_M = L - L_S)$.

(2) If $\min_{1\le j\le J}(B_{Scrl,j}) < B_S \le \max_{1\le j\le J}(B_{Scrl,j})$, we have $B_S \le B_{Scrl,j}$ for some classes (say, class j), and therefore $K_j(L_S>0, L_M=L-L_S) \le K_j(L_S=0, L_M=L)$. We have $B_{Scrl,i} \le B_S < B_{Scrh,i}$ for other classes (say, class i), and therefore $K_i(L_S > 0, L_M = L - L_S) > K_i(L_S = 0, L_M = L)$. Hence, the boundaries of $A(L_S > 0, L_M = L - L_S)$ and $A(L_S = 0, L_M = L)$ cross with each other.

(3) If $\max_{1\le j\le J}(B_{Scrl,j}) < B_S < \min_{1\le j\le J}(B_{Scrh,j})$, then $B_{Scrl,j} < B_S < B_{scrh,j}$ for $j = 1, ..., J$. Based on Proposition 5.1, we have $K_j(L_S > 0, L_M = L - L_S) > K_j(L_S = 0, L_M = L)$, for $j = 1, ..., J$. Therefore, $A(L_S = 0, L_M = L)$ is a proper subset of $A(L_S > 0, L_M = L - L_S)$.

(4) If $\min_{1\le j\le J}(B_{Scrh,j}) \le B_S < \max_{1\le j\le J}(B_{Scrh,j})$, we have $B_S \ge B_{Scrh,j}$ for some classes (say, class j), and therefore $K_j(L_S > 0, L_M = L - L_S) = K_j(L_S = 0, L_M = L)$; we have $B_{scrl,i} < B_S < B_{Scrh,i}$ for other classes (say, class i), and therefore $K_i(L_S > 0, L_M = L - L_S) > K_i(L_S = 0, L_M = L)$. Hence, $A(L_S > 0, L_M = L - L_S)$ and $A(L_S = 0, L_M = L)$ coincide at certain corner points and elsewhere $A(L_S > 0, L_M = L - L_S)$ subsumes $A(L_S = 0, L_M = L)$.

(5) If $B_S \ge \max_{1\le j\le J}(B_{Scrh,j})$, then we know from Proposition 5.1 that $K_j(L_S > 0, L_M = L - L_S) = K_j(L_S = 0, L_M = L)$ for $1 \le j \le J$. So, $A(L_S > 0, L_M = L - L_S) = A(L_S = 0, L_M = L)$.

Appendix D

Operation and Procedure for Chapter 6

D.1 Operation of the "Classical" Leaky Bucket

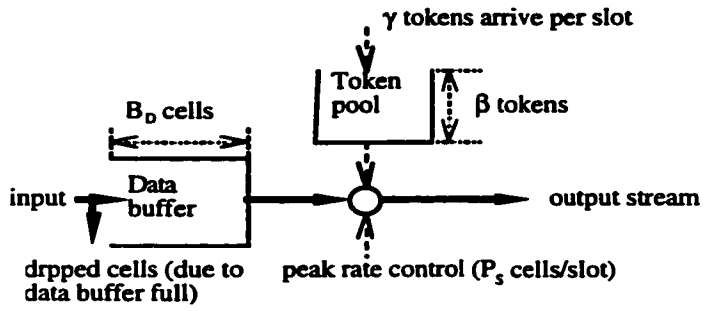


Fig. D-1 Illustration of the "Classical" Leaky Bucket Mechanism

The "classical" leaky-bucket (LB) mechanism was originally proposed to implement the traffic regulating / policing function at the network entrance to ensure that the entering traffic conforms to a certain traffic characterization. It usually has four control parameters (P_S, β, γ, B_D). These parameters are determined in the admission control process, so that the QoS can be guaranteed and network congestion is avoided. As shown in Fig. D-1, one token pool of size β cells and one data buffer of capacity B_D cells are available. Tokens are generated at a deterministic rate of γtokens per slot and stored in the pool if it is not full; otherwise, the newly generated token is lost. Similarly, a packet will be dropped if it arrives at a full data buffer, otherwise it will enter the FIFO queue at the data buffer. The packet at the head of the queue can be transmitted only when the token pool is not empty

and at least $1/P_S$ time-slots have passed since the last departure. In this "classical" LB mechanism, each packet transmission consumes one token from the token pool. Thus, the token arrival rate γ tokens/slot bounds the long-term average rate of the output traffic in cells/slot. Operating under this mechanism, the leaky bucket guarantees that the number of output cells during an arbitrary time interval of length t does not exceed the negotiated traffic constraint function $I^*(t) = min(P_S \cdot t, \ \gamma \cdot t + \beta)$, where P_S bounds the output stream's peak-rate measured in cells / slot; γ bounds its long-term average rate measured in cells / slot; β is its maximum burst size measured in cells.

D.2 Steps to Solve $\bar{\pi}$ from $\bar{\pi} = \bar{\pi} \bullet P$ for a M/G/1 Queue [Neut89]

Step 1: Find the matrix $\mathbf{G} \stackrel{def}{=} \sum_{v=0}^{\infty} A_v G^v$.

G can be obtained as the limit of the following matrix sequence $\{G_n\}_{n \geq 0}$:

$$\mathbf{G} = \lim_{n \to \infty} \mathbf{G}_n \quad .$$

where

$$\mathbf{G}_0 = \mathbf{0} \ ;$$

$$\mathbf{G}_n = \sum_{v=0}^{\infty} \mathbf{A}_v \mathbf{G}_n^v \qquad for \ n \geq 1 \quad .$$

Step 2: Find $\bar{\mathbf{g}}$, the probability vector associated with the matrix G.

Solve the linear equations: $\begin{cases} \bar{g} \cdot G = \bar{g} \\ \bar{g} \cdot \bar{e} = 1 \end{cases}$ to obtain $\bar{g}$.

Step 3: Obtain the matrix $K = \sum_{v=0}^{\infty} B_v G^v$.

Step 4: Find $\bar{k}$, the probability vector associated with the matrix K.

Solve the linear equations: $\begin{cases} \bar{k} \cdot K = \bar{k} \\ \bar{k} \cdot \bar{e} = 1 \end{cases}$ to obtain $\bar{k}$.

Step 5: Obtain vector $\bar{k}^*$.

$$A = \sum_{v=0}^{\infty} A_v ;$$

$$\bar{\beta} = \sum_{v=1}^{\infty} v \cdot A_v \cdot \bar{e} ;$$

$$\mu = (I - G + \bar{e} \cdot \bar{g})[I - A + (\bar{e} - \bar{\beta}) \cdot \bar{g}]^{-1} \cdot \bar{e} ;$$

$$\bar{k}^* = K \cdot \bar{e} + \sum_{n=1}^{\infty} B_n \sum_{k=0}^{n-1} G^k \mu \quad .$$

Step 6: Find π_0.

$$\pi_0 = \frac{\bar{k}}{\bar{k} \cdot \bar{k}^*} \quad .$$

Step 7: Find π_j.

$$A_j^* = \sum_{k=j}^{\infty} A_k G^{k-j} \quad ;$$

$$B_j^* = \sum_{k=j}^{\infty} B_k G^{k-j} \quad ;$$

$$\pi_j = [\pi_0 B_j^* + \sum_{k=1}^{j-1} \pi_k A_{j+1-k}] \cdot (I - A_1^*)^{-1} \quad .$$

Appendix E

OPNET Simulation

We present the OPNET simulation models used in Chapter 6 to investigate the impact of rate-controlling on end-to-end performance. OPNET requires the queuing models to be set up in three levels: network, node, and process models.

E.1 Network Model

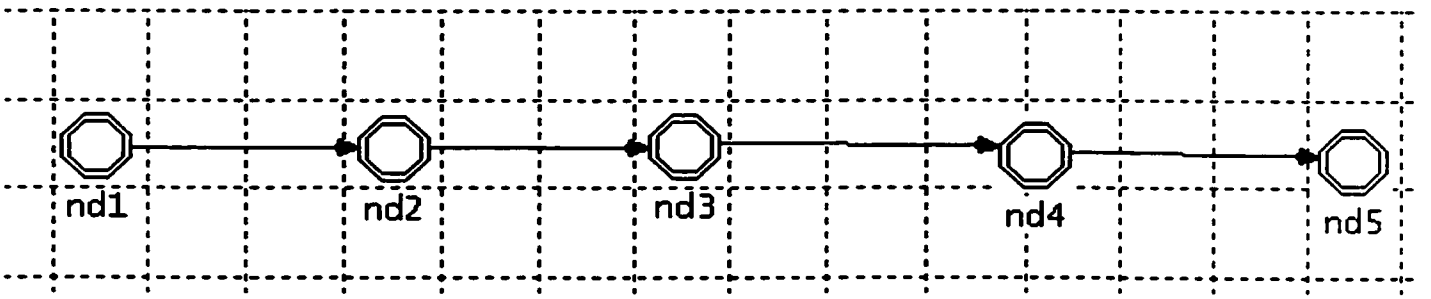


Fig. E-1 The Network Model of Simulation

Fig. E-1 is the network model of the simulation in Chapter 6. There are five nodes in this tandem network: the source node nd1, three intermediate nodes, nd2, nd3, and nd4, and the destination node nd5. Their node models will be discussed in the next section.

E.2 Node Models

Three node models have been set up, i.e., the node model of the source node, node model of the intermediate node, and the node model of the destination node. They are depicted in the following Fig. E-2, E-3, and E-4, respectively.

In Fig. E-2, the traffic generator module *g_0* generates the reference traffic by a specified inter-arrival time process.

In figures E-2, E-3, and E-4, the processor module *RC* models the dual-leaky-bucket rate-controller. The processor module *q_0* models the FIFO traffic multiplexer. The transmitter module *xmt* transmits traffic to the next node. The receiver module *rcv* receives traffic from the upstream node. The processor modules *p_1* to *p_10* are background traffic generators, each of which generates one background traffic stream.

In Fig. E-4, the processor module *sink* models a drain for the packets, i.e., it destroys all the received packets.

The details of the processes will be discussed in the next section.

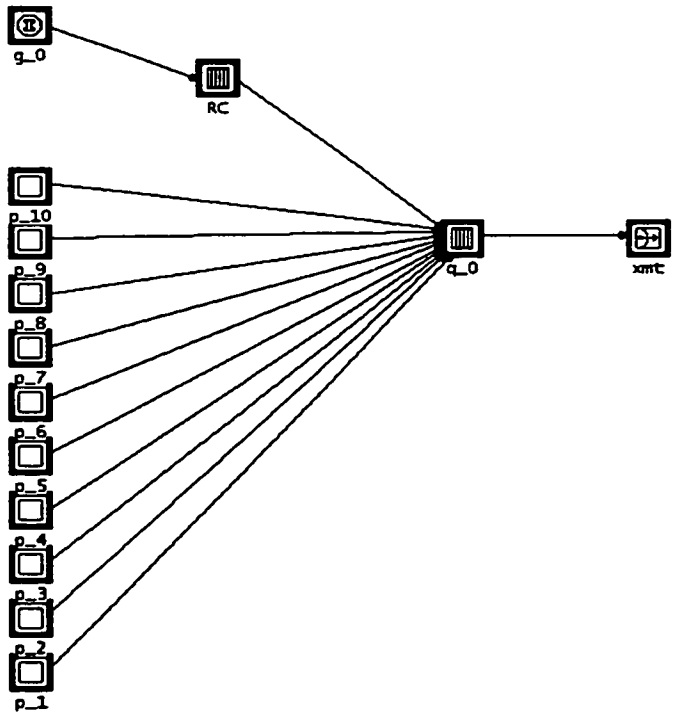


Fig. E-2 Node Model of the Source Node

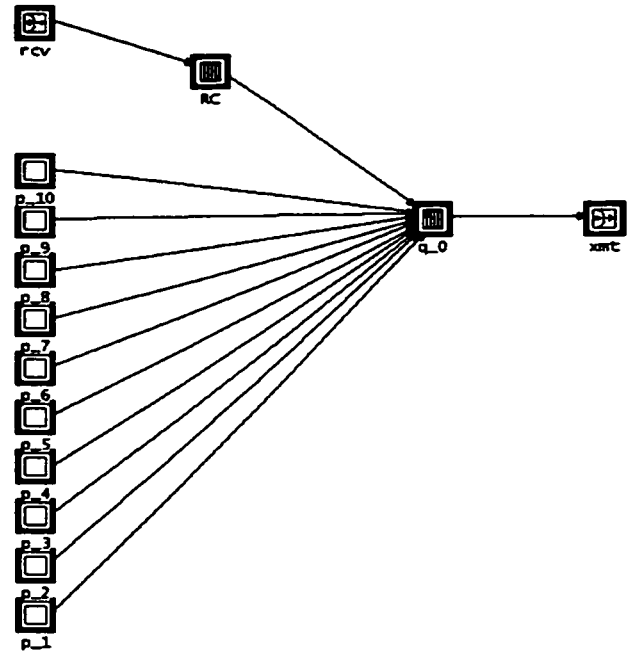


Fig. E-3 Node Model of the Intermediate Node

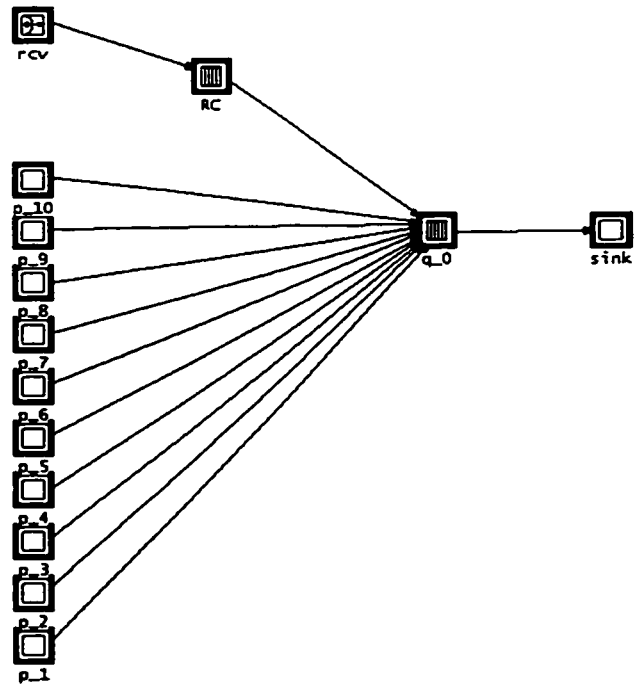


Fig. E-4 Node Model of the Destination Node

Description of the reference traffic generator module *g_0*

The reference traffic generator module *g_0* is not implemented by a process model. Instead, it is configured at the nodal level by setting the attribute values (Fig. E-5).

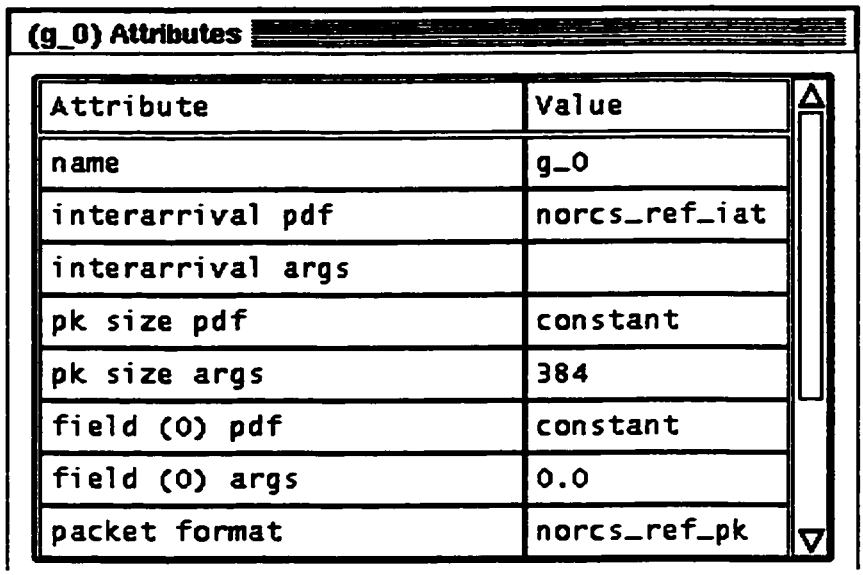


Fig. E-5 Description of the Reference Traffic Generator Module

In Fig. E-5, we see that all the packets generated by the module g_0 have a constant packet size 384 bits. The packets are generated according to the inter-arrival time process *norcs_ref_iat* (Fig. E-6), which is a typical inter-arrival time process of a VBR traffic flow. The packet format will be discussed in the next section.

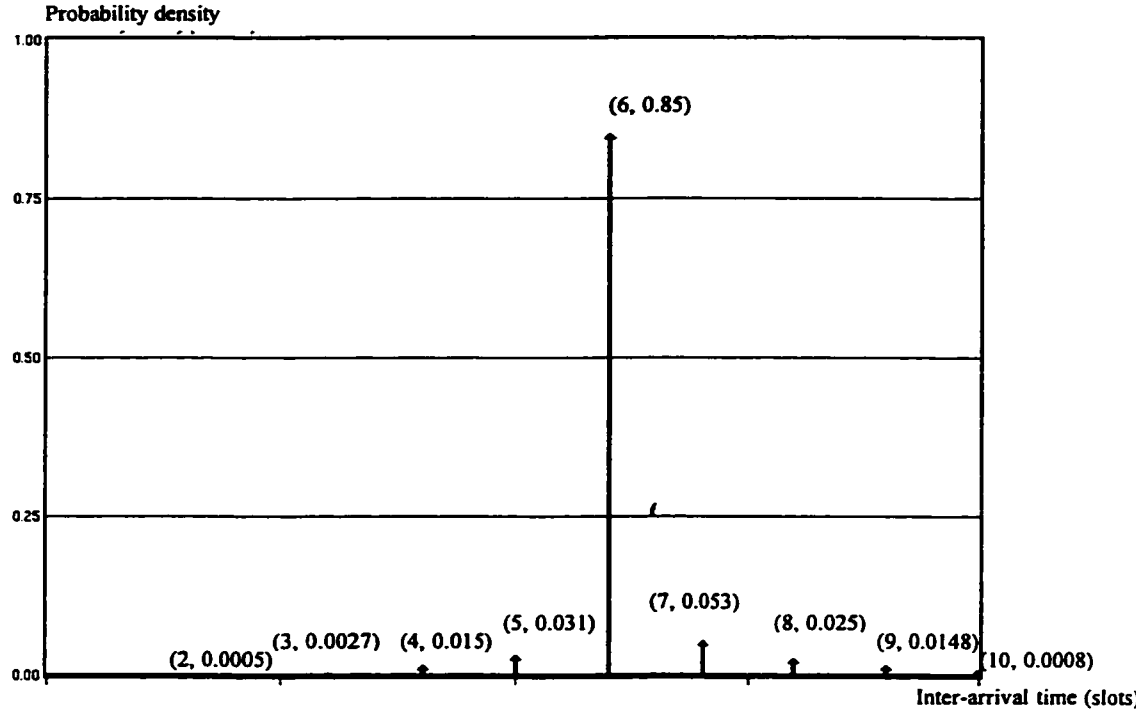


Fig. E-6 The Inter-arrival Time Process of the Reference Traffic

E.3 Packet Models

Two packet models have been set up. They are the reference packet model *norcs_ref_pk* and the background packet model *norcs_bkgd_pk*. Both packet models have one sub-field "src_addr", and the size of this sub-field is zero. The value of this sub-field is set to 0 in the *norcs_bkgd_pk*, and it is set to 1 in the *norcs_ref_pk*.

The packet size for both packet models is set to 384 bits (48 bytes), which is the payload length in an ATM cell. According to the ATM Forum, the ATM cell size is 53 bytes. However, the cell size varies in the real ATM products. In our simulation, because we only care about the traffic control aspect instead of the protocol aspect of an ATM network, we do not use ATM header, and set the cell size as 48 bytes (384 bits). Also, we set the output link capacity being 384 bits per time-slot. That is, the service rate of a link is 1 cell /slot.

E.4 Process Models

Three process models have been set up, i.e., the process model of the background traffic generator *norcs_bsrc_proc*, the process model of the rate-controller *VBR_shp*, and the process model of the FIFO traffic multiplexer *rcstest1_q_proc*.

Fig. E-7 presents the state transition diagram of the process model for the background traffic generator. The functions of each state are listed as follows.

(1) The state "Init" initializes the variables.

(2) The state "Control" generates a uniformly distributed variable between 0 and 1 every time slot, and controls the probability of sending out a packet during one time-slot as *avg_rate*.

(3) The state "wr" computes the average load of the generated background traffic.

Fig. E-8 is the state transitional diagram of the process model for the FIFO traffic multiplexer. The functions of each state are listed as follows.

(1) The state "Init" initializes variables and registers the statistics.

(2) The state "arrival" inserts the received packets into the FIFO queue, and computes the inter-arrival time for the reference stream.

(3) The state "svc_start" starts serving the packet at the head of the FIFO queue, and schedules the time to finish the service.

(4) The state "svc_compl" sends out the packet, records the time of the current departure from the reference stream, and collects the statistics for the reference stream, including the end-to-end delay, the delay at multiplexer, the node delay, and the inter-departure time.

(5) The state "wr" computes and outputs the statistics of the reference stream, including the average and variance of the end-to-end delay, the node delay, and the delay at the multiplexer. It also computes the probability distributions of the inter-departure time, the inter-arrival time, the delay at the multiplexer, the end-to-end delay, and the node delay.

Fig. E-9 gives the state transitional diagram of the process model for the rate-controller. The functions of each state are listed as follows.

(1) The state "Init" initializes the variables and registers the statistics.

(2) The state "arrvl" inserts the new arrival from the reference stream into the queue, and schedules the transmitting time of the new arrival according to the rules of our proposed leaky-bucket mechanism (section 6.2.1).

(3) The state "send" sends out the packet at the head of the queue, records the time of the current departure, and computes the inter-departure time.

(4) The state "tkngen" generates a slot-token every time-slot.

(5) The function of state "wr" computes the statistics for the reference stream, including the mean and variance of the shaping delay, the probability distributions of the shaping delay, the inter-departure time, and the inter-arrival time.

The C-programming software codes for these process models are summarized in the next section.

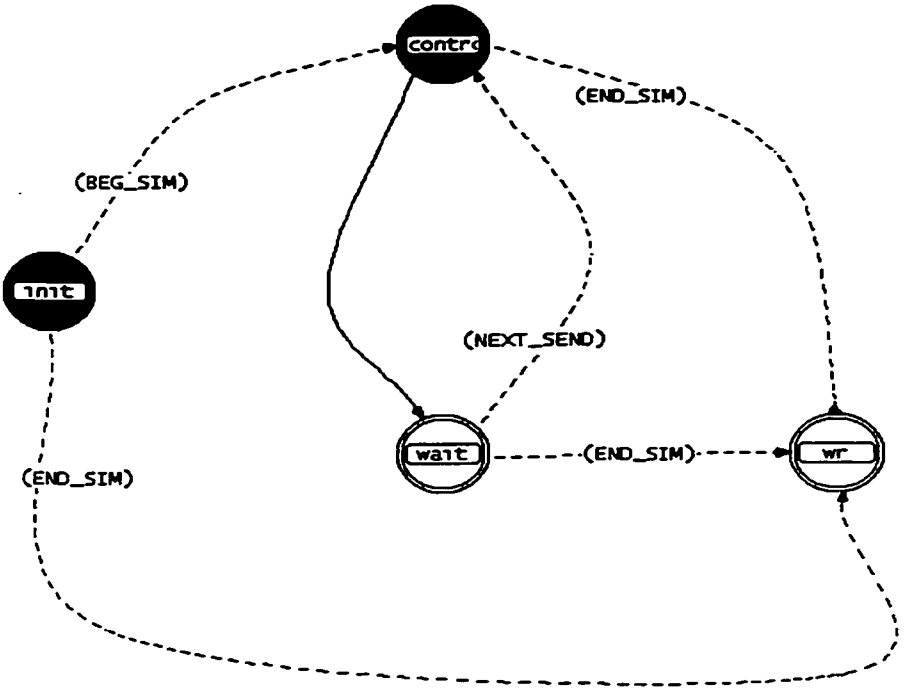


Fig. E-7 Process Model of the Background Traffic Generator

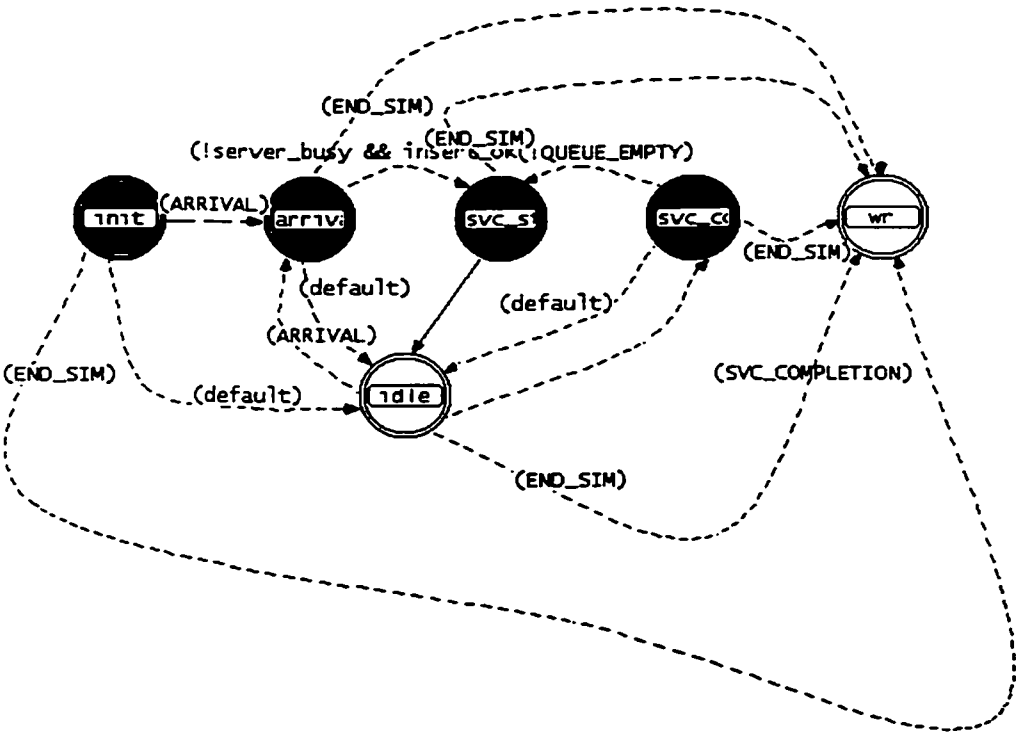


Fig. E-8 Process Model of the FIFO Traffic Multiplexer

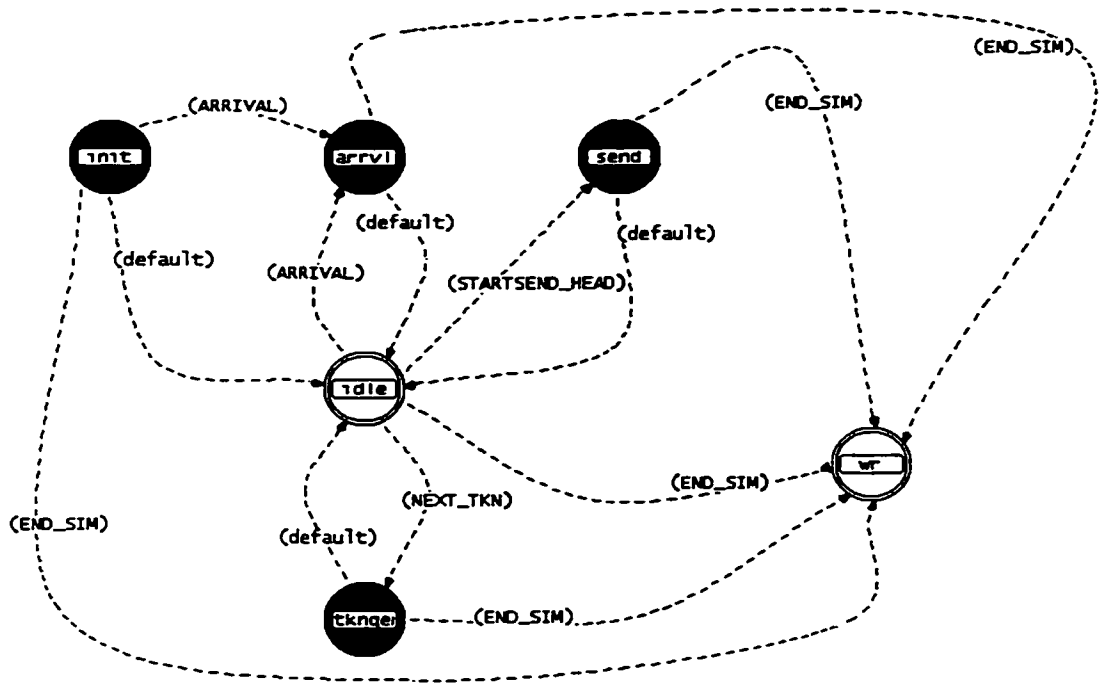


Fig. E-9 Process Model of the Leaky Bucket Rate-controller

E.5 Program Code

OPNET uses a C-type language along with their OPNET kernel codes [Mil97]. The functions of each process in section E.4 are provided in the following.

Module Name	Functions
norcs_bsrc_proc	generates packets according to a Bernoulli arrival process;
rcstest1_q_proc	simulates a FIFO queue at a traffic multiplexer;
VBR_shp	implements the proposed leaky bucket mechanism.

The programming codes are not attached here due to the agreement with MITEL. But it is available upon request from our CCNR report.