PAPER Special Issue on Telecommunications Network Planning and Design

A Delayed Vacation Model of an M/G/1 Queue with Setup Time and Its Application to SVCC-Based ATM Networks

Mahbub HASSAN† *and* Mohammed ATIQUZZAMAN†, *Nonmembers*

SUMMARY The emerging ATM network will support permanent, semi-permanent and switched virtual channel connections (SVCC). A number of simulation studies have been carried out to study the performance of SVCCs using empirical data. The absence of an analytical model has prevented the study of SVCCs under known traffic distributions. In this paper, we develop a new *delayed vacation* model to facilitate the performance study of SVCCs with configurable inactivity timer in the ATM network interface card (NIC). Comparison with simulation results indicates that the proposed model is very accurate and can be effectively used to optimise the performance of SVCCs by selecting an appropriate inactivity timer.

key words: *ATM, switched VCC (SVCC), queueing theory*

1. Introduction

Asynchronous Transfer Mode (ATM) is a *connection-oriented* network. An end-to-end connection, called a virtual channel connection (VCC), has to be setup first before information can be exchanged between two end-systems. Two different VCC setup mechanisms have been proposed [1]: (i) *Permanent VCC (PVCC)*, where a VCC is setup *a priori* and left open indefinitely and (ii) *Switched VCC (SVCC)*, where a VCC is setup and closed down dynamically as needed using a *signalling protocol*.

The advantage of PVCC is that no time is wasted in setting up a VCC when there is data to transmit. With PVCC, however, some network resources (e.g. bandwidth) are wasted when the VCC is not in use and since a PVCC is required for every pair of end-systems connected to an ATM network, communication using PVCC is not scalable to large networks.

With SVCC, waste of resources and the number of concurrently open VCCs in the network are minimised by dynamically opening and closing a VCC. However, setting up a VCC using a signalling protocol involves some *delay* for the data to be transmitted, *processing overhead* for the end-systems and the intermediate switches and *extra signalling traffic* in the network.

One way to reduce the cost of SVCC is to reduce the number of VCC setups by implementing an *inactivity timer* to manage the closing of an inactive or idle VCC. An idle VCC is closed down only if no data arrive within the timeout interval. This reduces the number of VCC setups at the price of some wasted resource between the time the VCC becomes idle and is closed down. Consequently, the length of the timer plays a crucial role in the amount of resource waste and the savings due to reduced number of VCC setups.

The goal of this paper is to develop an analytical model to analyse the operation of an SVCC and obtain various performance measures as a function of the inactivity timer value. We model an SVCC with a queue and a server which goes into vacation when inactivity timer runs out. We propose a *delayed vacation model* of an M/G/1 queue with setup time to perform the analysis. The accuracy of the model is verified through simulation of a widely used data communication application (TELNET) over an SVCC connection. The simulation results closely match the results obtained from the analytical model.

The paper is organised as follows. Section 2 summarises (i) previous work by other researchers on the inactivity timer of SVCC and (ii) the queueing models available in the literature that has some relevance to our study. The analysis of our proposed model is presented in Sect. 3 with its application to the performance analysis of SVCC in Sect. 4. The simulation model we have used to validate our proposed *delayed vacation model* is presented in Sect. 5 and the results of the simulation are provided in Sect. 6. Finally, conclusions are made in Sect. 7.

2. Previous Work

2.1 SVCC Analysis

Saran and Keshav [2] studied two *pricing schemes* for a VCC and two *holding time policies* of an idle VCC using empirical dataset of packet arrivals. In the first pricing scheme, there is no cost associated with idling of an open VCC, but the total number of VCCs that a terminal can open is restricted. However, since open VCCs will reserve some resources in an ATM network, a second pricing scheme (with no restriction on the number of open VCCs) was studied which includes a cost per time unit for an idle VCC.

The first holding policy (called Least Recently

Manuscript received July 5, 1996.
Manuscript revised September 10, 1996.

†The authors are with the Monash University, Australia.

Used (LRU)) studied in [2] exploits the temporal locality of packet arrival times. They used a linear monotonic function to predict the next arrival time. The second holding policy calculates a new timeout value each time a VCC becomes idle. The new timeout value is based on a simple estimation technique of the mean and standard deviation of the packet interarrival times at each packet arrival.

Later, Lund et al. [3] and Keshav et al. [4] have shown that better performance can be achieved with adaptive timers where the timeout is calculated based on the distribution of the previous packet interarrival times. It should be stressed that the above policies are all based on very high capacity VCCs, and hence *queueing of packets at the VCC was not considered.*

The authors of [2]–[4] have not analysed the performance of SVCCs as a function of timeout values for static timers. Clark et al. [5] have studied SVCCs as a function of timeout values for static inactivity timers by *simulating* various ATM configurations. Since the simulation was conducted on data collected from an existing network, the results are very specific to a particular network and *fails to provide general results* for known traffic distributions.

2.2 Queueing Models

Standard queueing models which assume that the server is always up and running can effectively model a PVCC where the VCC is always open. Researchers in the past [6] have studied *N-policy with setup time* models where the server goes to a *vacation* after serving the last packet in the queue. The server comes back from vacation when N packets arrive, and then spends a *setup* (or *startup*) time in preservice work prior to serving packets. The vacation and setup time are analogous to closing the VCC and VCC setup time in a SVCC environment. However, the previous models [6] can only be applied to the case where the VCC is closed *immediately* after transmitting the last packet in the queue. The effect of the inactivity timer is not captured in such models. Our proposed model (see Sect. 3) effectively models the inactivity timer in the SVCC by *delaying* a vacation by some time. To the best of our knowledge, our proposed model is the first one to analyse an ATM SVCC.

3. Delayed Vacation Model

In this section we develop and analyse our proposed *delayed vacation model* of an M/G/1 queue with setup time. The application of this queueing model to the analysis of an ATM SVCC will be shown in Sect. 4.

3.1 Model Description

Our proposed queueing model is shown in Fig. 1. There is an infinite queue associated with a single server. The

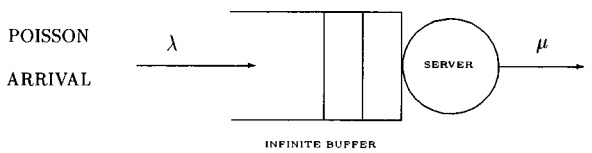


Fig. 1 M/G/1 queueing model.

arrival process to the queue is Poisson with a mean arrival rate of λ. The service time has a general distribution with an average of $\frac{1}{\mu}$. The packets are served in the order first-in-first-out (FIFO).

The server serves all packets in the queue exhaustively until the queue becomes empty. After serving the last packet in the queue, the server waits T time units anticipating the arrival of more packets. If no packet arrives within this T time units then the server is turned off† to conserve resources such as bandwidth. If the server is off when a packet arrives, it is turned on immediately. However, there is a service setup period of C time units for the server. In other words, the server takes C time units from the moment it is turned on to the beginning of the service of the packet(s) waiting in the queue. The server will be in any one of the following four *states* at any instant of time:

SERVE: The server is serving a packet.

WAIT: Although the queue is empty, the server is on, ready to serve and waiting for a packet to arrive.

OFF: The server is turned off (no packet in queue).

SETUP: The server has been turned on by a packet arrival but is not yet ready to serve; the server is doing some preservice work.

The transitions among the above four states are shown in Fig. 2. It is worth noticing that for $T = 0$, (i.e., if the server is turned off immediately after serving the last customer in the queue), the server goes to OFF state directly from the SERVE state without entering the WAIT state at all. Therefore, for $T = 0$ the server has only three states and the model reduces to a special case of an *M/G/1 system under N-policy with general setup time* studied in [6].

3.2 Model Analysis

In this section, we analyse the *delayed vacation model* proposed in the previous section. We are interested in estimating the following quantities, as a function of the known parameters λ, μ, C and T, which will enable us to analyse the performance of an SVCC to be discussed in Sect. 4:

†The concept is similar to going to a vacation. Instead of going to a vacation immediately after serving the last customer in the queue, a vacation (if there is one) is delayed by T time units. Hence the name *delayed vacation model.*

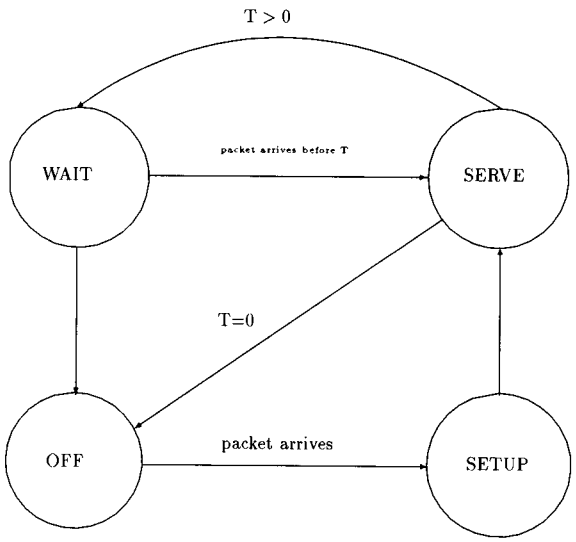


Fig. 2 State transition diagram for the delayed vacation model.

1. Setup rate γ or the average number of times the server enters the SETUP state per unit time.
2. Average delay W per packet (time spent in queue plus the service time).
3. Average number of packets $\mathcal{N}$ in the system.
4. Fraction of time the server spends in the different states (WAIT, SERVE, OFF and SETUP).

3.2.1 Setup rate γ

The setup rate γ can be calculated by converting our delayed vacation model into an M/G/1 system under 1-policy (c.f. N-policy in [6]) with general setup time. We can then make use of some available results found in [6].

The system goes through two main *cycles*: (i) *busy* cycle serving customers (SERVE) or doing preservice work (SETUP) and (ii) *idle* cycle (OFF or WAIT) not doing any work. We define the *idle cycle rate* (β) as the number of idle cycles per unit time and is calculated as:

$$\beta = \frac{\text{Fraction of time spent in idle cycles}}{\text{Average length of an idle cycle}} \tag{1}$$

The numerator and denominator of the above equation for an M/G/1 system under 1-policy with general setup times are given by $\frac{1-\rho}{1+\lambda\bar{C}}$ and $\frac{1}{\lambda}$ respectively [6] where $\bar{C}$ is the expected setup time and $\rho = \lambda/\mu$ is the *utilisation* of the server. Since every busy cycle is preceded by an idle cycle, the rate of busy cycles is equal to the idle cycle rate β.

For $T = 0$, every busy cycle starts with a setup of C time units. Hence the setup rate $\gamma = \beta$ and the expected setup time $\bar{C} = C$. Therefore, β for $T = 0$ is given by:

$$\beta|_{T=0} = \frac{\lambda(1-\rho)}{1+\lambda C} \tag{2}$$

For the delayed vacation model ($T > 0$) some busy cycles start with a SETUP state while the rest do not. The SETUP states can therefore, be classified into two types. One type of SETUP requires zero setup time and the other type requires C time units. SETUP states with zero setup times are preceded by idle cycles of length $I \leqq T$ which has a probability of $1-e^{-\lambda T}$ for exponential packet interarrival times. Similarly, SETUP states with C units of setup time has a probability of $e^{-\lambda T}$. Therefore, the expected setup time for the *delayed vacation model* is $\bar{C} = Ce^{-\lambda T}$. The idle cycle rate for the delayed vacation model ($T > 0$) is thus given by:

$$\beta|_{T>0} = \frac{\lambda(1-\rho)}{1+\lambda Ce^{-\lambda T}} \tag{3}$$

For $T > 0$, only idle cycles of duration $I > T$ will be followed by a SETUP state. Therefore, the setup rate γ is given by

$$\gamma = \Pr[I > T]\beta|_{T>0} = \frac{e^{-\lambda T}\lambda(1-\rho)}{1+\lambda Ce^{-\lambda T}} \tag{4}$$

The correctness of Eq. (4) is verified by the fact that $\lim_{T\to\infty}\gamma = 0$.

3.2.2 State Probabilities

Let π_s, π_v, π_o and π_w denote the probabilities of the server being in the SETUP, SERVE, OFF and WAIT states respectively. Probability of being in a state is equivalent to the fraction of time spent in that state. Therefore, from Eq. (4) we get,

$$\pi_s = \frac{Ce^{-\lambda T}\lambda(1-\rho)}{1+\lambda Ce^{-\lambda T}} \tag{5}$$

From [6] we get,

$$\pi_v + \pi_s = \rho + \frac{C\lambda e^{-\lambda T}(1-\rho)}{1+\lambda Ce^{-\lambda T}} \tag{6}$$

From Eqs. (5) and (6) we get,

$$\pi_v = \rho \tag{7}$$

Since the fraction of time spent in the SERVE state represents the utilisation of the server, we observe that the utilisation of the server for the delayed vacation model remains $\rho = \frac{\lambda}{\mu}$ and is not a function of setup (C) or delay (T) times.

The system goes through a busy cycle followed by an idle cycle followed by a busy cycle and so on as mentioned in the previous section. Some of the idle cycles include an OFF state while others do not. An idle cycle includes an OFF state if the duration of the cycle is greater than T. If an idle cycle includes an OFF state then it must have spent time T in the WAIT state before going to the OFF state. If an idle cycle does not include an OFF state, then it spends time $I \leqq T$ in the WAIT state (before going to the SERVE state), where

the length of I is exponentially distributed between 0 and T. The probability that an idle cycle includes an OFF state is $e^{-\lambda T}$. Since π_w is the fraction of time spent in the WAIT state, we get

$$\pi_w = e^{-\lambda T}\beta|_{T>0}T + \beta|_{T>0}E(I)|_{I\leqq T} \tag{8}$$

where $E(I)|_{I\leqq T}$ is the expected value of I and can be obtained as $\int_0^T I\text{pdf}(I)dI$. Since I is exponentially distributed, we get

$$E(I)|_{I\leqq T} = \int_0^T I\lambda e^{-\lambda I}dI = \frac{1}{\lambda}[1 - e^{-\lambda T}(\lambda T + 1)] \tag{9}$$

Since all the state probabilities should add to 1, we can obtain π_o as:

$$\pi_o = 1 - \pi_s - \pi_v - \pi_w. \tag{10}$$

3.2.3 Average Delay W

From *P-K* formula [7] the average waiting time (W) of a customer in a M/G/1 system is obtained as:

$$W = \bar{X} + \frac{\lambda\bar{X^2}}{2(1-\rho)} \tag{11}$$

where $\bar{X}$ and $\bar{X^2}$ are the first and second moments of the service time and ρ is the utilisation of the server. For the delayed vacation model, it has been shown in Sect. 3.2.2 that $\rho = \frac{\lambda}{\mu}$. Therefore,

$$W = \bar{X} + \frac{\lambda\bar{X^2}}{2(1-\frac{\lambda}{\mu})} \tag{12}$$

Applying *Little's formula* the average number of customers $\mathcal{N}$ in the system is obtained as

$$\mathcal{N} = \lambda\bar{X} + \frac{\lambda^2\bar{X^2}}{2(1-\frac{\lambda}{\mu})} \tag{13}$$

$\bar{X}$ and $\bar{X^2}$ of the delayed vacation system can be calculated as follows. The packets are classified into two categories: (i) *normal* packets which arrive to the system when the server is in WAIT, SERVE or SETUP state and (ii) *exceptional* packets which arrive when the server is in OFF state. The average service time for normal packets is $\bar{X}_\text{n} = \frac{1}{\mu}$ and the average service time for the exceptional packets is $\bar{X}_\text{e} = \frac{1}{\mu} + C$. If P_e is the probability that a packet is an exceptional packet, we can derive the first and second moments as:

$$\bar{X} = P_\text{e}\bar{X}_\text{e} + (1-P_\text{e})\bar{X}_\text{n} = \frac{1}{\mu} + P_\text{e}C \tag{14}$$

$$\bar{X^2} = P_\text{e}\bar{X}_\text{e}^2 + (1-P_\text{e})\bar{X}_\text{n}^2 \tag{15}$$

where $\bar{X}_\text{e}^2$ and $\bar{X}_\text{n}^2$ are the second moments of the service times of the exceptional and normal packets respectively.

Note that P_e is equivalent to the fraction of packets which require setups which is simply $\frac{\gamma}{\lambda}$. Substituting $P_\text{e} = \frac{\gamma}{\lambda}$ in Eqs. (14) and (15) we obtain

$$\bar{X} = \frac{1}{\mu} + \frac{\gamma}{\lambda}C \tag{16}$$

$$\bar{X^2} = \frac{\gamma}{\lambda}\bar{X}_e^2 + \left(1 - \frac{\gamma}{\lambda}\right)\bar{X}_\text{n}^2 \tag{17}$$

Calculation of the second moment depends on the distribution of service time. As an example, for determinstic service time, $\bar{X}_\text{e}^2 = (\frac{1}{\mu} + C)^2$, $\bar{X}_\text{n}^2 = (\frac{1}{\mu})^2$ and the second moment can be obtained from Eq. (17) as:

$$\bar{X^2} = \frac{1}{\mu^2} + \frac{\gamma C^2}{\lambda} + \frac{2\gamma C}{\mu\lambda} \tag{18}$$

For exponential service time, $\bar{X}_\text{e}^2 = 2(\frac{1}{\mu} + C)^2$, $\bar{X}_\text{n}^2 = 2(\frac{1}{\mu})^2$ and the second moment is obtained as:

$$\bar{X^2} = 2\left(\frac{1}{\mu^2} + \frac{\gamma C^2}{\lambda} + \frac{2\gamma C}{\mu\lambda}\right) \tag{19}$$

The second moment of other distributions can be calculated in a similar fashion.

4. Delayed Vacation Model and SVCC

In this section, we demonstrate that the proposed *delayed vacation model* described in the preceding section can be effectively applied to model and study a static timer based ATM SVCC as shown in Fig. 3. The performance measures of the SVCC model are described below:

- *Setup rate* γ: is measured as the average number of VCC setup per unit time.
- *Average delay* W: is measured from the time a packet is generated for transmission over the VCC until it is completely transmitted.
- *Idling ratio* ζ: is measured as the fraction of time an open VCC is idling on average. This can be obtained from the state probabilities of *the delayed vacation model* as $\frac{\pi_w}{\pi_w+\pi_v}$.

The correspondence between the parameters of the *delayed vacation model* and the SVCC model are summarised in Table 1. To verify the effectiveness of the delayed vacation model in modeling an ATM SVCC, we have developed a simulation model and compared the results obtained from the simulation and the proposed model which are presented in the next two sections.

5. Simulation Model

TELNET is a terminal emulator application supported

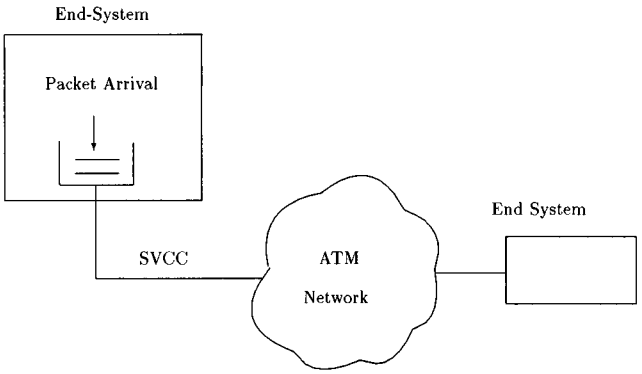


Fig. 3 A Model of an ATM SVCC.

Table 1 Correspondence between the parameters of the SVCC and the M/G/1 delayed vacation model.

M/G/1 Parameter	*SVCC Parameter*
vacation delay time T	length of inactivity timer T
service setup rate γ	VCC setup rate γ
packet arrival rate λ	packet arrival rate λ
average service time $\frac{1}{\mu}$	packet tx time on the VCC

by TCP (Transmission Control Protocol) in the Internet. We have developed a simulation model of a TELNET connection in an ATM LAN (local area network) as shown in Fig. 4 and discussed below. TCP receives packets (each keystroke results in one packet) from TELNET. The arrival of packets to TCP can be modeled with a Poisson process as the packets are generated by a human being at a keyboard. TCP sends packets from the send-buffer to the VCC under the constraint of a window flow control mechanism which relys on acknowledgment from the receiving TCP entity. TCP maintains a retransmit-timer for each packet sent. If the timer expires before the acknowledgement for a packet is received, the packet is retransmitted. The propagation time in the ATM network (LAN) is considered negligible.

We assume that the typed characters are echoed locally. The sending end-system, therefore, does not generate ACK packets. We also assume that the receiving TCP generates an ACK immediately upon receiving a packet. We simulate very fine grain TCP retransmit timers. The values of the TCP parameters used in our simulations are shown in Table 2. A detailed description of these parameters can be found in [8].

Various parameters of the ATM connection are shown in Table 3. With one-byte TELNET packets, the packet generation rate (for 4bps) is $\lambda = \frac{4}{8} = 0.5$ packets/sec which is the arrival rate to the TCP. Since every one-byte TELNET packet results in two 53-byte ATM cells (due to TCP, IP and AAL5 [9] overhead), a VCC of 4 kbps has a speed of $\mu = 4.72$ packets/sec.

We have run many simulations (with varying timer length T) to observe various SVCC performance measures as a function of T. The results of our simulation along with a comparison with results obtained from our proposed model are presented next.

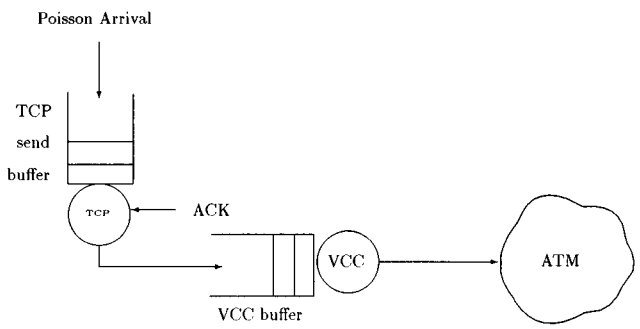


Fig. 4 Separate VCC model for each TCP connection.

Table 2 TCP/IP parameters for simulation experiment.

parameter	*value*
TCP maximum window size	64 Kbyte
TCP header	20 bytes
IP header	20 bytes
minimum TCP retx-timeout	0.03 sec
maximum TCP retx-timeout	10 sec
maximum retransmission retry	16

Table 3 ATM parameters for the simulation experiment.

parameter	*value*
VC tranmission rate	4 Kbps
mean arrival rate from TELNET	4 bps
VC setup delay	50 ms
ATM adaptation layer	AAL5
Cell loss rate in ATM network	negligible

6. Results

We have run each simulation for 10 hours and 10 minutes of simulated time. The statistics are collected after a warmup period of 10 minutes to allow the system to reach a steady state. The VCC setup rate (γ) and VCC idling ratio (ζ) obtained from simulations are compared with those obtained from the proposed analytical model in Figs. 5 and 6. A very close match between simulation and analysis results shows the accuracy of the model proposed in this paper.

The average delay (W) in the system for a TELNET packet is measured as the sum of average delay in the TCP buffer, average delay in the VCC buffer and the transmission time of the two ATM cells. Figure 7 shows W as a function of T. The analytical results are within 2% of simulation results for higher values of T ($T > 2$ sec) and are exactly the same for lower values of T. We have investigated the reason for the difference in the case of higher values of T and found that it is due to the timing out of TCP retransmit timer. We found that although there is no loss in the system, the TCP timer used in the simulations often resulted in some re-

transmissions for large values of the SVCC inactivity timer.

To show the effect of the TCP retransmit timer on the average packet delay W, the retransmission rate of TCP has been overlayed on W in Fig. 7. It can be seen that the delay obtained from simulation starts to shift from the values obtained from the analytical model when TCP retransmission rates increase. This happens when the SVCC inactivity timer is greater than 2 sec. after which the higher retransmission of packets increases the number of packets in the VCC buffer which in turn increases the delay.

To confirm that the discrepancy between the delay values obtained from the simulation and the model are actually due to the TCP retransmit timer, we have repeated the simulations with the TCP retransmit timer turned off. Since there is no loss in the system, TCP still functions properly without any retransmissions. The delay results are shown in Fig. 8. As can be seen, the results from the simulation exactly match the ones from the analysis.

While our model may provide a slightly optimistic result (less than 2%) in calculating W for certain values of the inactivity timer for retransmission based protocols (e.g. TCP), it will provide exact results for protocols which do not use retransmit timers (e.g. UDP).

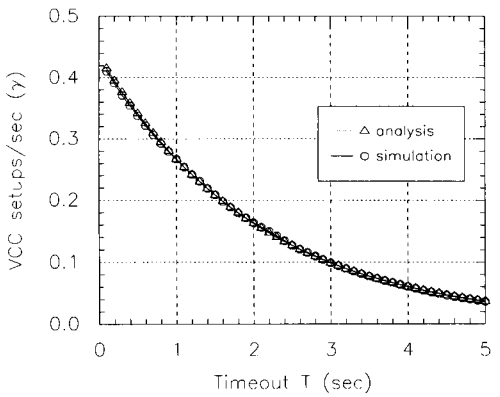


Fig. 5 VCC setup rate as a function of inactivity timer.

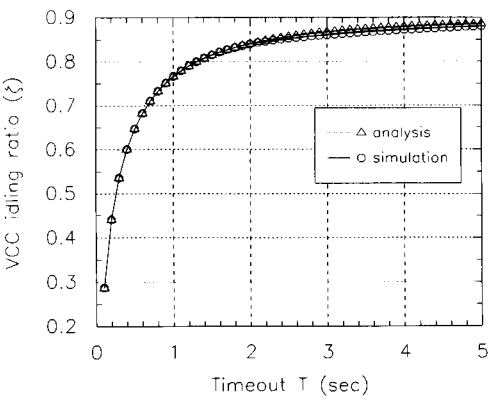


Fig. 6 VCC idling ratio as a function of inactivity timer.

7. Conclusion

We have developed a new *delayed vacation* model to study the performance of an SVCC-based ATM network. The model can effectively account for the in-

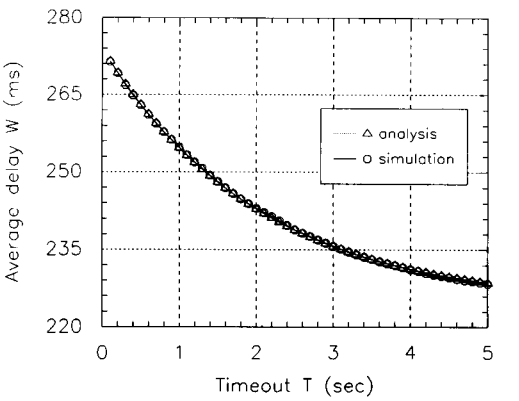


Fig. 8 W as a function of T, with TCP timer turned off.

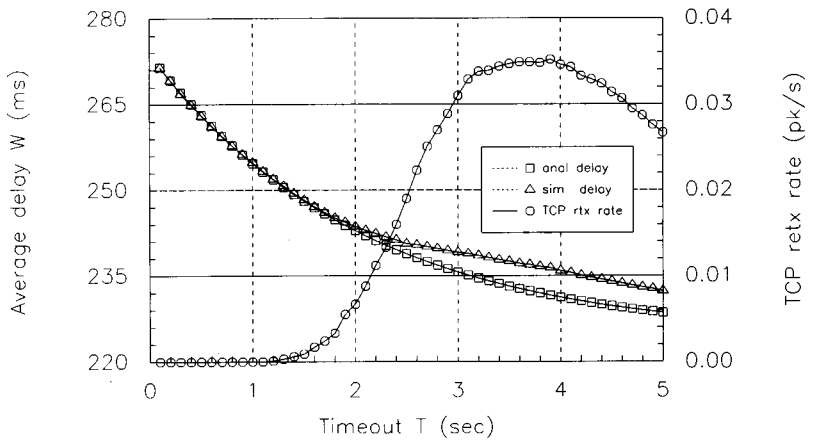


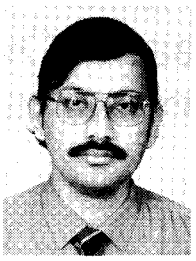
Fig. 7 Delay in the system as a function of inactivity timer.

activity timer in the Network Interface Card (NIC) at a terminal or end-system connected to an ATM network. The model has been validated by comparison with simulation results of a TELNET session over an ATM SVCC. The model will allow users to configure the timer value in NICs to optimise the performance of SVCCs.

References

[1] ITU-TS, "Rec. I.311: B-ISDN General Network Aspects," ITU, Geneva, March 1993.
[2] H. Saran and S. Keshav, "An empirical evaluation of virtual circuit holding times in IP-over-ATM networks," in INFOCOM, pp.1132–1140, 1994.
[3] C. Lund, S. Phillips, and N. Reingold, "Adaptive holding policies for IP over ATM networks," in INFOCOM, pp.80–87, 1995.
[4] S. Keshav et al. "An empirical evaluation of virtual circuit holding policies in IP-over-ATM networks," IEEE J. Select. Areas Commun, vol.13, no.8, pp.1371–1382, Oct. 1995.
[5] R.J. Clark, R.R. Hutchins, and S.W. Register, "Deploying ATM in a data network: An analysis of SVC requirements," Technical Report, College of Computing, GIT, Atlanta, GA 30332-0280, USA, March 1995.
[6] J. Medhi, "Stochastic Models in Queueing Theory," pp.297–415, Academic Press, 1991.
[7] D. Bertsekas and R. Gallager, "Data Network," Prentice Hall, 2nd Edition, 1992.
[8] D.E. Comer and D.L. Stevens, "Internetworking With TCP/IP," vol.2. Prentice Hall, 2nd Edition, 1994.
[9] ITU-TS, "Rec. I.363: B-ISDN Adaptation Layer (AAL) Specification," ITU, Geneva, Sect. 6, Jan. 1993.

Mahbub Hassan is an academic staff member in the Gippsland School of Computing and Information Technology at Monash University, Australia. He received the B.Sc. degree in Computer Engineering from the Middle East Technical University, Turkey in 1989 and the M.Sc. degree in Computer Science from the University of Victoria, Canada in 1991. He is now at the final stage of the Ph.D. degree in Digital Communications at Monash University, Australia. He has served as a Session Chair in the International Technical Conference on Computers, Circuits/Systems and Communications held in Seoul in 1996. He is a member of IEEE. His current research interests are in the performance of data communications over ATM Networks and LAN interconnection through Broadband ISDN. He has published and reviewed many technical papers in the above areas. He can be contacted at Mahbub. Hassan@fcit. monash. edu. au.



Mohammed Atiquzzaman received the M.Sc. and Ph.D. degrees in electrical engineering and electronics from the University of Manchester Institute of Science and Technology, England in 1984 and 1987 respectively. He had been academic staff member at La Trobe University, Melbourne and King Fahd University of Petroleum & Minerals, Saudi Arabia. Currently he is an academic staff member in the department of Electrical & Computer Systems Engineering at Monash University, Melbourne, Australia. He serves on the editorial board of the IEEE Communications Magazine and has been guest editors of special issues on "ATM Switching" and "ATM Networks" of the International Journal of Computer Systems Science & Engineering. He has also served as technical program committee member of several international conferences including IEEE INFOCOM. His current research interests are in Broadband ISDN and ATM networks, multiprocessor systems, interconnection networks, parallel processing and image processing. He has published widely in the above areas. He can be contacted at atiq@eng. monash. edu. au.