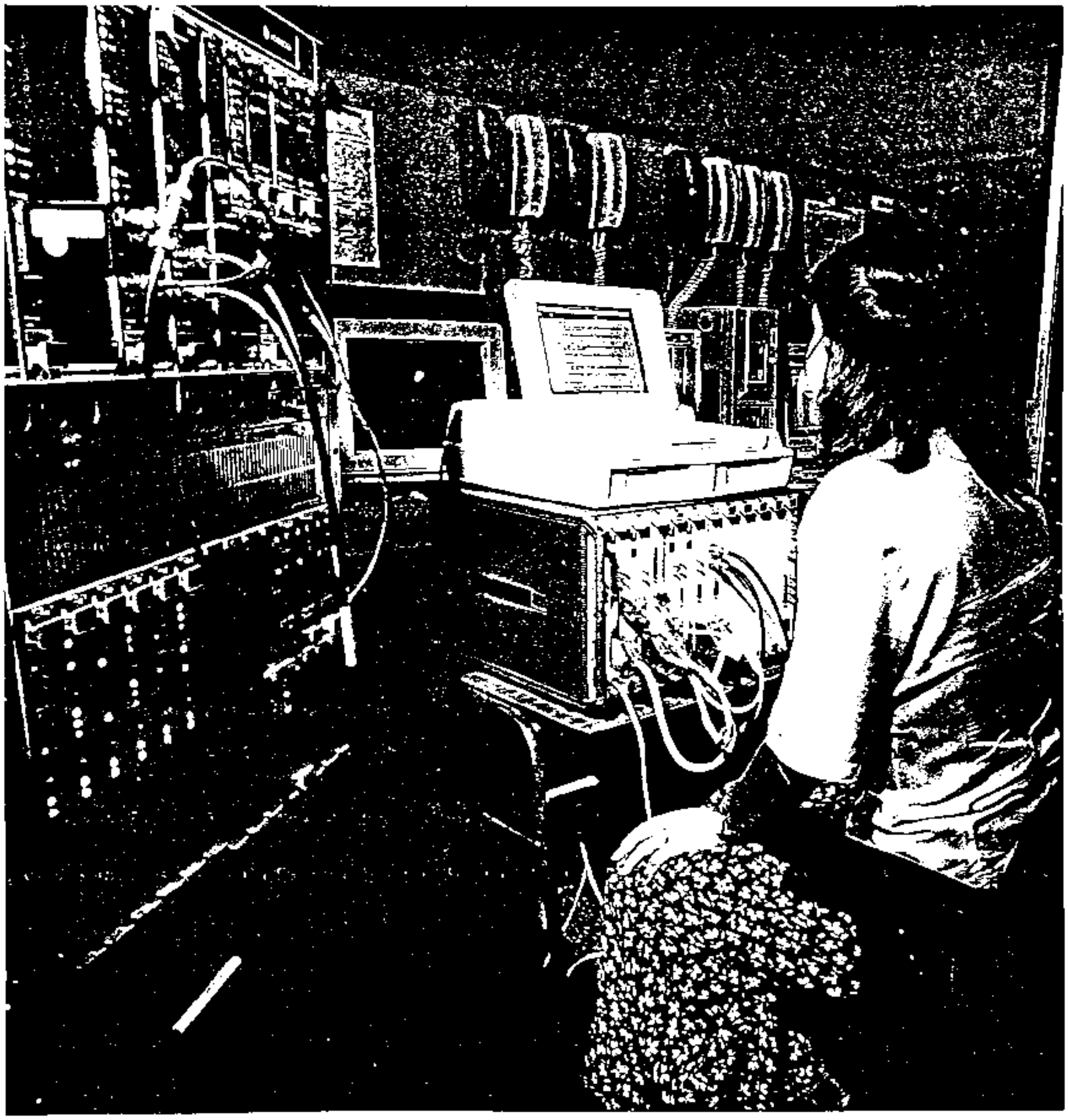
Proceedings of Australian Telecommunication Networks & Applications Conference 1996

"Delivering Quality Services"



Melbourne, Australia

3 - 6 December 1996

Data Communications over ATM using SVCC: A Survey of Issues, Problems and Solutions

Mahbub Hassan
School of Computing & Information Technology
Monash University Gippsland Campus
Churchill VIC 3842, Australia
Mahbub.Hassan@fcit.monash.edu.au

Mohammed Atiquzzaman
Dept. of Electrical & Computer Systems Eng.
Monash University, Clayton Campus
Clayton VIC 3168, Australia
atiq@eng.monash.edu.au

Abstract

Connections in an ATM network are made using virtual channel connections which can be either permanent (PVCC) or switched (SVCC). SVCC requires a signalling protocol which is being standardised by international standards organisations. Because of the cost effectiveness of SVCCs over PVCCs for data communication in an ATM environment, a significant amount of work has been carried out to determine the most cost-efficient and effective ways to operate SVCCs. These include issues such as holding time policies, pricing schemes, effect of call blocking, etc. The aim of this paper is to survey the work done in the area of SVC with special emphasis to the above mentioned issues.

1 INTRODUCTION

Various standardisation activities (e.g. [10, 6, 7] are currently in progress to enable transportation of existing of data communications applications over future Asynchronous Transfer Mode (ATM) networks. Existing data communications protocols are based on *connection-less* concept; routing is performed on every data frame based on the destination address contained in the header of each frame. Since ATM is designed as a *connection-oriented* network, one of the main issues in supporting data communications over ATM is the design of mechanisms to carry connection-less traffic over connection-oriented network.

International Telecommunications Union (ITU) has standardised a connection-less service [16] where the user of the service does not have to establish an end-to-end connection. This is achieved by building a network of connection-less servers (CLS) within the ATM network. The user terminals are connected to one of these CLSs through an ATM connection. A CLS is capable of routing each data frame separately based on the destination address contained in the frame header, assembling and reassembling of frames into ATM cells, and switching ATM cells to appropriate ATM VCs to forward the cells to the next CLS or to the destination user terminal. This type of connection-less support over ATM is also referred to as *direct* [21] or *server-based* [5] method.

While the server-based method is simple and may be attractive to the users because of the lack of end-to-end connection establishment at the user terminal, maintaining a network of CLSs is expensive for the ATM network provider which will probably result in an extra charge for the users of this service. The alternative to using the server-based connection less service is implementing end-to-end connection management techniques at the user terminal. Such techniques are also referred to as *indirect* or *connection-based* methods and are similar to existing data communications over connection oriented data networks, e.g., IP-over-X.25 [2].

An end-to-end connection between two user terminals connected to an ATM network is called an ATM virtual channel connection (VCC). A permanent or semi-permanent VCC (PVCC) may be established *a priori* between every pair of terminals. The advantage of using PVCs is the elimination of the use of signalling protocol and associated VCC setup delay. Connection management at the user terminal involves simply a table lookup to determine the correct virtual circuit identifier (VCI) depending on the destination address of the data frame.

PVCs, however, require a mesh connection among all the user terminals. In fact, there may be more than one PVCs required between a pair of terminals if different applications running on the same terminal require different quality of service (QoS). Hence PVCs do not scale well with large networks.

In order to alleviate the scalability problem, Chao et al. [4] have proposed a PVC-based approach which maintains a number of PVCs between ATM ports in an ATM switch instead of PVCCs between every pair of user terminals. A central connection manager maintains a *free-list* and an *allocated-list* for all the PVCs. A terminal has to exchange a set of query-response messages with the central manager inorder to obtain a PVC. Although this method reduces the total number PVCs, it introduces some delay through negotiation with the central connection manager. This method also does not scale well with large ATM switches with many ports and hence is not suitable for wide area ATM network.

An end-to-end connection can also be setup dynamically as needed through the use of switched VCC (SVCC). The user terminals use a signalling protocol (e.g. [17]) in order to setup a VCC with the destination terminal whenever one is needed. The benefits of transporting connecless-less data over SVCC is twofold. Firstly, the scalability problem is greatly reduced as a SVCC can be closed when not in use. Secondly, closing down an idle VCC eliminates any resource waste associated with an idling open VCC. SVCC will be, therefore, one of the popular methods for data communications over ATM.

The purpose of this paper is to summarise the issues regarding data communications over ATM using SVCC with a focus to identify the problems.

The rest of the paper is organised as follows. In Section 2, we examine the timing issue of closing an idle open VCC and its impact on the total cost of data communications. The performance of file transfer using a widely used data communications protocol — Transport Control Protocol (TCP) — over ATM using SVCC is reported in the next section. Section 4 describes analytical and simulation methods reported in the literature to measure various performance parameters of data communications over SVCC-based ATM networks.

2 HOLDING POLICIES

The main idea of using SVCC instead of PVCC is to close a VCC when it is not needed any longer. The simplest way to manage SVCCs is to open a SVCC when a data frame arrives and close it after the last cell of the frame has been transmitted over the VCC [19]. However, using this method, every data frame will experience the VCC setup delay which both degrades the performance of data communications and creates excessive signalling traffic in the ATM network. Since each setup might attarct an additional charge, a frame-by-frame setup method might be too expensive.

If after transmitting a frame it is expected that more frames will arrive soon, then it may be wise to keep the VCC open for a while. This will amortise the cost VCC setups over many frames. A VCC *holding policy* refers to the policy which decides on how long to hold an idle VCC. The aim of such holding policies is to minimise the total cost based on some cost model.

To make a decision on how long to hold the idle VCC open using some holding policy involves some processing time and resource. Therefore, the less often the decision has to be made the better. For very high speed ATM connections, a frame will be transmitted before the next one arrives. Such high speed connections were studied in [20, 18, 9] and holding decisions were made when the VCC becomes idle after transmitting each frame. A slow VCC model, where the frame arrival rate in a burst could be greater than the VCC bandwidth, was studied in [13]. In that study, the arriving frames are buffered in a FIFO queue while a VCC is being setup or is busy transmitting a frame. The holding decisions for such slow VCC model with buffer have to be made only when the buffer becomes empty.

Most of the holding policies proposed in the literature make use of an inactivity timer [7]. A timer is set when a VCC becomes idle or the buffer becomes empty. The VCC continues to be open even if there is no data to send. If no frame arrives within the timeout period, the timer goes off and the VCC is closed. If a frame arrives before the timer expires, the timer is cleared (it will not go off); a new timer is set when the VCC becomes idle again. One holding policy differs from the other on how to select the timeout interval of the inactivity timer.

Based on the selection of timeout, the timer-based holding policies can be broadly classified into two groups:

- Fixed timer. The same timeout is used each time a timer is set. This is very simple and implemented by the ATM adapter card vendors.

- Adaptive timer. The timeout is recalculated for every timer based on the past frame interarrival times.

The selection of the timeout for both fixed and adaptive timers wil depend on the underlying cost model. Since ATM is a connection oriented network and will allocate some resource to an open VCC, a time-based charge in addition to a setup charge is reasonable (like the existing telecommunications network). For such cost model, there are two main cost factor — setup cost and holding time cost.

Saran and Keshav [20] studied holding policies with both fixed and adaptive timers using a time-based cost model. The study was based on empirical data collected from different LANs. One of the holding policies, referred to as LRU-based (Least Recently Used), is based on the temporal locality of packet arrival times in the empirical data sets. The underlying theory of this policy is that the predicted arrival time of the next packet since the last arrival to an idling VCC increases monotonically with elapsed time.

Saran and Keshav [20] uses a linear monotonic function which predicts the next arrival time to be a positive constant times the elapsed time since the last arrival. For the empirical data sets, it was found that a positive constant of 2 gives the best cost performance. Therefore, the LRU-based holding policy is simply to close a VCC if it has been idle for $\frac{S}{2}$ time units where S is the ratio of VC setup cost to VC holding cost. Note that for LRU-based policy, a fixed timeout is used for every timer and can be easily implemented with the available commercial ATM adapter cards. However, one disadvantage of the LRU-based policy is that it will not work well if the temporal locality does not exist.

Saran and Keshav [20] suggested another policy where a new timeout is calculated each time a VCC becomes idle. Estimation of mean μ and standard deviation σ of the packet interarrival times are updated at each packet arrival using a simple estimation technique. Upon becoming idle, the VCC is closed if $\mu \geq S$, otherwise a timer is set with a timeout equal to $\max(\mu+2\sigma, \frac{S}{5})$. Although temporal locality is not important for this policy, it has more overhead in updating and maintaining estimation of mean and deviation and resetting a timeout for each timer. It was shown that LRU-based policy performed better than this policy for the empirical data sets.

Later, Lund et. al. [18] and Keshav et al [9] have shown that better performance can be achieved with adaptive timers where the timeout is calculated based on the distribution of the packet interarrival times. The arrival distribution is approximated by maintaining a histogram of the last 10-20 arrival times. However, this method requires storage of some past data for each VC and is more complex computationally.

The above policies are all based on very high capacity VCCs. It was assumed that the time it takes to process a data packet (segmentation into cells and transmission of cells over the VCC) is less than the packet interarrival times. VC setup delay was also considered negligible. Therefore, queueing of packets at the VC was not considered.

Even with very high capacity VCC, some

buffer is required for non-negligible VCC setup delay to hold the arriving frames while a VCC is being setup. It was argued in [1] that with small buffers, there may be some loss of data and therefore, frame loss should be considered as a cost factor. Based on the new cost model and an ON-OFF traffic arrival model it was shown through simulation [1] that fixed timer based holding policies can not optimise the total cost. The least cost is achieved either by closing the VCC immediately after buffer becoming empty or leaving it open permanently depending on the cost associated with a VCC setup, frame loss or unit of holding time.

Later, an analysis of slow VCC model (during a burst frames arrive faster than the VCC can process and transmit) with infinite buffer [11] confirmed that for ON-OFF traffc model, fixed timers can not optimise cost for a time-based cost model. It is shown in [11] that the optimum holding policy is to close the VCC immediately on buffer becoming empty if the holding cost for the mean silence period is greater than the setup cost, leave it open otherwise.

3 TCP PERFORMANCE

In this section, results of performance studies of TCP over ATM networks are reported for various kinds of data applications.

3.1 Bulk Data Transfer

Most reliable transport layer data communications protocols, including TCP, implement some kind of *sliding window* flow control mechanism. With sliding window, the transport layer can transmit a number of data frames (equal to the window size) without receiving any acknowledgement from the receiver. For sufficiently large window, an acknowledgement will be received before the entire window is transmitted and the transport entity will be able to transmit continuously without having to stop for acknowledgement. Therefore, for large windows the VCC will not be idle in the middle of a file transfer and hence the performance of SVCC will be the same as the PVCC.

However, if the window size is not large enough to fill the pipe of a large delay-bandwidth end-to-end path, the transport entity will stop after transmitting a window full of frames and the VCC will get idle. If VCC gets idle, there is a possibility that an inactivity timer will go off and the VCC is closed. The performance of bulk TCP connections for a simple holding policy — the VCC is closed immediately upon becoming idle — is analysed in [15] for such small window cases. It was shown that the relative throughput loss (c.f. PVCC) for TCP due to VCC setup delay could be significant in the case where the ratio of VCC setup delay to the round-trip propagation delay is large.

3.2 Small File Transfer

TCP implements a *dynamic* window size mechanism as a preventive measure against congestion in the network. When a new TCP connection starts or a new file transfer starts on a TCP connection which has been idle for a while, the window size is initially set to one and incremented by one for each packet acknowledged. This is called slow-start [3]. After the window size reaches a threshold, the window size is incremented by one only after all the packets of the previous window are acknowledged (congestion-avoidance [3]). If a TCP packet is lost, the TCP retransmit timer will go off and TCP will enter the slow-start phase. For bulk TCP analysis in [15], the TCP performance was analysed at the steady state with no loss, i.e., when the window size reaches the maximum size. For steady-state and loss-less environment, TCP analysis is the same as a *fixed* window size sliding window protocol.

A performance analysis of dynamic window size TCP conections over SVCC is presented in [14]. A number of TCP connections were simulated multiplexed onto the same SVCC managed by a fixed inactivity timer. It was found that TCP performance can be severely degraded if a timeout less than the round-trip time is selected for the fixed inacativity timer of the SVCC connection.

3.3 Interactive Data

The performance of interactive data transfer (e.g. TELNET) is not measured by throughput but by the delay each individual data frame experiences. If the interframe gap is large compared to the VCC transmission rate, it is possible that the VCC buffer will become empty often and the possibility that a frame will have to experience a VCC setup delay will be very high. Therefore, it will be necessary to multiplex a number of such low rate connections onto the same VCC whenever it is possible to do so.

3.4 Effect of Call Blocking

With SVCC, a connection request may be rejected by the connection admission control (CAC) functions of the ATM network if there is not enough resources in the network to meet the quality of service of the connection. This is somewhat analogous to the collission scenario of Ethernet LAN. Upon detecting a collision, terminals connected to an Ethernet LAN wait a random time before trying to send the frame again. The random durations for successive retries (exponential backoff) are standardised in IEEE 802.3. However, the retry techniques are not yet standardised for SVCC-based ATM networks. The analyses of SVCC in [11, 12, 14, 1, 13, 15, 8] are all based on negligible call blocking probability. Further study is necessary to evaluate the performance of SVCC under non-negligible call blocking probability.

4 MODELING AND ANALYSIS

When using a fixed timeout based inactivity timer to manage a SVCC, selection of the timeout plays a crucial role in determining a number of parameters — VCC setup rate, number of concurrent VCCs in the network, fraction of time a VCC is idling, average delay of a data frame in the VCC buffer, the average size of the VCC buffer and so on. Clark et al. [8] have studied the VCC setup rate and the number of concurrent VCCs as a function of the timeout of the inactivity timer by simulating various ATM configurations to support an existing campus network. The study showed that the VCC setup rate and the number of concurrent VCCs increases rapidly if a timeout less than 90 second is selected.

An analytical model of a SVCC was developed in [12] and analysed to obtain the VCC setup rate, SVCC idling fraction, average delay of a data frame and the average queue size for a Poisson traffic arrival model and a fixed timeout based inactivity timer to manage the SVCC. Simulation of TELNET data over SVCC verified the accuracy of the proposed model.

5 CONCLUSION

This paper has summarised the work that has been carried out in different issues relating to the operation of the SVC. There is now a need to integrate the isolated studies to obtain a feeling of the overall performance of SVCCs. It has been observed that most of the work has been carried out using simulation. Some of the authors have reported results of simulation studies using emperical data collected from a specific network configuration. Although these results provide useful insight into the performance of the operation of the SVCC, it is necessay to develop models which would enable one to study the performance of SVCCs under known traffic patterns.

References

[1] M. Atiquzzaman and M. Hassan. Effect of Frame Loss on the Holding Time of a Virtual Circuit in ATM LAN Interconnection. In *19th Australasian Computer Science Conference*, pages 81–86, Melbourne, January/February 1996.

[2] J.J. Barret and E.F. Wunderlich. LAN Interconnection Using X.25 Network Services. *IEEE Network Magazine*, pages 12–16, September 1991.

[3] Douglas E. Comer and David L. Stevens. *Internetworking With TCP/IP, Vol. 2.* Prentice Hall, 2nd edition, 1994.

[4] H.J. Chao et al. IP on ATM Local Area Networks. *IEEE Communications Magazine*, pages 52–59, August 1994.

[5] J. Boudec et al. Connectionless Data Service in an ATM-based customer premises Network. *Computer Networks and ISDN Systems*, 26:1409–1424, 1994.

[6] M. Laubach et al. *RFC1577:Classical IP and ARP over ATM.* Network Working Group, January 1994.

[7] M. Perez et al. *Internet Draft draft-ietf-ipatm-sig-01.txt: ATM signalling support for IP over ATM.* IP over ATM WG, July 1994.

[8] R.J. Clark et al. Deploying ATM in a Data Network: An Analysis of SVC Requirements. Technical report, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, email:rjc@cc.gatech.edu, March 1995.

[9] S. Keshav et al. An Empirical Evaluation of Virtual Circuit Holding Policies in IP-Over-ATM Networks. *IEEE JSAC*, 13(8):1371–1382, October 1995.

[10] LAN Emulation SWG Drafting Group. *94-0035R2+:LAN Emulation over ATM, Draft Specification - Revision 2.* ATM Forum, April 1994.

[11] M. Hassan. In Search for an Optimum Inactivity Timer for Data Communications over ATM Networks. In *International Technical Conference on Circuits/Systems, Computers and Communications*, pages 393–396, Seoul, July 1996.

[12] M. Hassan and M. Atiquzzaman. A Delayed Vacation Model of an M/G/1 Queue with Setup time and its Application to SVC-based ATM Networks. Submitted to IEICE Transactions on Communications.

[13] M. Hassan and M. Atiquzzaman. Loss and Delay Study of LAN Interconnection over ATM. In *Australian Telecommunication Networks & Applications Conference*, pages 185–190, Sydney, December 1995.

[14] M. Hassan and J.W. Breen. A Study of TCP Performance Over ATM with Switched Virtual Circuits. In *IEEE Singapore International Conference on Communication Systems*, Singapore, November 1996.

[15] M. Hassan and J.W. Breen. Calculation of Throughput Loss Due to ATM Connection Setup Time. In *IASTED International Conference*, pages 129–132, Orlando, Florida, January 1996.

[16] ITU-TS. *Recommendation I.364: Support of Broadband Connectionless Data Service on B-ISDN.* ITU, Geneva, January 1993.

[17] ITU-TS. *Recommendation Q.2931: B-ISDN Digital Subscriber Signalling System No.2 (DSS2), User Network Interface layer 3 Specification for basic call.connection control.* ITU, Geneva, June 1994.

[18] C. Lund, S. Phillips, and N. Reingold. Adaptive Holding Ploicies for IP over ATM Networks. In *INFOCOM*, pages 80–87, 1995.

[19] Masayuki Murata and Hideo Miyahara. LAN Internetworking Through BISDN. *IEICE Trans. Comm.*, E77-B(3):294–305, March 1994.

[20] H. Saran and S. Keshav. An Empirical Evaluation of Virtual Circuit Holding Times in IP-over-ATM Networks. In *INFOCOM*, pages 1132–1140, 1994.

[21] B.J. Vickers and T. Suda. Connectionless Service for Public ATM Networks. *IEEE Communications Magazine*, pages 34–42, August 1994.