Experimental Results from a Practical Implementation of a Measurement Based CAC Algorithm. Contract ML704589 Final report

Andrew Moore and Simon Crosby

May 1998

Abstract

Interest in Connection Admission Control (CAC) algorithms stems from the need for a network user and a network provider to forge an agreement on the Quality of Service (QoS) for a connection the user wishes to have admitted into the network. The CAC algorithm must balance the competing interests of the user wishing to obtain a desired QoS from the new connection and the need of a network provider to maintain the agreed QoS of existing connections. This study uses the Fairisle ATM LAN integrated into a test-rig specifically designed to compare CAC algorithms to assess a threshold-based CAC algorithm for ATM networks. Two sets of results are shown; the first for a preliminary study used to establish parameters of the CAC as well as parameters of the evaluation experiments. The second set of results covers the evaluation of the CAC algorithm and includes a comparison with the results gained using other CAC algorithms and results from theoretical models.

Results in this study show that threshold values calculated by BTL collaborators were optimistic giving a higher than desired CLR value. The relationships between CLR, mean line utilisation and threshold were established empirically for a variety of sources and network connection load conditions. These results enabled a comparison between traffic types that possessed the same broad traffic descriptors such as Peak Cell Rate, showing that the threshold values depended not only on connection load but also on traffic type. This study presents the empirically established results for the relationships between CLR, mean line utilisation and threshold for a range of offered load, thereby allowing the effect of offered load on these relationships to be demonstrated. The concluding set of results of this report are the comparison of estimated results using the threshold mechanism with results for two other CAC algorithms.

Contents

List of Figures

List of Tables

1 Introduction

Connection Admission Control (CAC) denotes the set of actions taken by the network during the call set-up phase in order to accept or reject an ATM connection. A connection request is only accepted when sufficient resources are available to carry the new connection through the network at its requested Quality of Service (QoS) while maintaining the agreed QoS of existing connections. During the connection set-up phase the following information has to be declared, negotiated and agreed between the "user" and "network" to enable CAC to make a reliable connection acceptance/rejection decision:

- A Service Category (such as Constant Bit Rate (CBR) or non-real-time-Variable Bit Rate (nrt-VBR))
- a QoS class expressed in terms of cell transfer delay, delay jitter and cell loss ratio (CLR), and
- specific limits on traffic volume the network is expected to carry.

For a given connection across a network it is not necessary that all these aspects of the CAC decision be declared every time. Many of the parameters, such as (CLR) or cell delay transfer may be implicit for the network on which the connection is being requested. As a result the actual information declared between a prospective "user" and a "network" may be only the service category and a characterisation of the traffic volume limits. An example of this would be that a call is made on the assumption a certain QoS is available in a network and when it makes its connection it declares that it is a nrt-VBR call with a specific Peak Cell Rate (PCR), Sustained Cell Rate (SCR) and Maximum Burst Size (MBS).

It can be seen that the algorithm controlling the decision made during the CAC will control the policy of the network. This decision will attempt to balance the requirements of the "user" (achieve the desired QoS) versus the requirements of the "network" (do not violate the QoS guarantees made to other pre-existing connections). As a result of this balance of "trade-offs", a highly pessimistic CAC algorithm may always achieve the QoS by assuming the worst possible characteristics about a new connection. As a result this algorithm would allow few connections into the network; in return for always achieving the QoS commitment, such a CAC algorithm would potentially waste resources – leaving much of the network under utilised. In comparison, a highly optimistic algorithm could always assume the best possible characteristics about a new connection. Such an algorithm would risk violating the QoS contracts made to existing connections for the sake of making maximal use of the available resources. An ideal CAC algorithm will achieve an even balance between "user" and "network".

During the development of such ideal CAC algorithms, substantial effort has been invested in modelling and experimenting with the entire network situation. Models are made of all aspects of the situation including traffic sources, network behaviour, the multiplexing of new connections and the variety of CAC algorithms available. However modelling

alone does not satisfactorily assess the behaviour of real CAC algorithms implemented in real situations. Additionally, common modelling techniques involve the use of simulated sources of traffic; these are used because they are well understood and easily generated. However due to the variety of sources and the continual development of new network users (and thus new types of traffic sources), such simulated sources of traffic do not represent adequately the range of behaviours such traffic sources can have in a network.

The inability to model the whole of a real-world CAC process and the inability to adequately represent real sources of traffic mean it becomes desirable to evaluate CAC algorithms in a controlled experimental situation using real traffic sources in an actual network. Through the use of modelled traffic sources a comparison can be conducted between the theoretical models of the CAC situation and the implementation; this enables both the feedback to improve CAC algorithms and a way to ensure realistic assumptions are made in the construction of CAC algorithms. By using real sources of traffic, such as video data streams or client-server file system traffic, the CAC algorithm can be tested against traffic sources that are less easily modelled thereby ensuring their usefulness in the real-world environment.

Section 2 discusses the sources used in this CAC evaluation. The first source type is based upon a theoretical model; this means it is both easily simulated and has well understood characteristics. The second source type has been created from video stream data and represents a common real-world network application. Section 2 discusses issues involved in the encoding of video for transport over a network.

Following on from the discussion of sources used in this evaluation, Section 3 details the function of CAC algorithms based upon thresholding techniques and how these algorithms adapt to the varying of network conditions. This section covers the concepts of "offered load" and the role played by this concept in the normalisation of differing network parameters in order to make threshold techniques more manageable. The BT adaptive CAC algorithm, also referred to as the Key CAC algorithm, is discussed and parameters for the evaluation of the CAC algorithm are given. This section concludes by documenting aspects of time scales as they relate to the functioning of the Key CAC algorithm.

Section 4 describes the experimental configuration used to conduct an initial set of experiments. The initial experiments were designed to establish parameters of the CAC mechanism as well as parameters of the evaluation experiments to be conducted. To this end the initial experiments consisted of simple configurations of equipment to source and sink traffic sources with traffic streams passing through the instrumented ATM switch. Section 4 makes a discussion of the unique characteristics of the ATM switch as well as the methods used to create multiple traffic streams for the initial experiments.

The objectives of the initial experiments documented in Section 5 were to give results that would enable our BTL collaborators to check effective bandwidth calculations and to investigate the relationship between CLR and buffer overflow. Additionally the initial experiments allowed the level of accuracy required for each experiment to be established; this established directly the minimum number of cells to give a balance between the run time of experiments and obtaining statistically significant values for CLR.

Section 6 describes the experimental configuration used in the evaluation of the BT adaptive Threshold based CAC algorithm. The philosophy in the design and construction of the CAC test-rig was to allow comparison of one CAC algorithm with another under near identical conditions of connection load. Such a requirement implies that the evaluation environment must allow control of traffic types and connection characteristics such as attempt rate and connection holding time. In addition to controls of connection load, the test-rig must allow substitution of one CAC algorithm for another while allowing information about the performance of the system to be collected and compared. Such needs lead to three goals for the test-rig; high performance – allowing connection rate, adaptable – allowing multiple CAC algorithms each with its own requirements for measurement and calculation and finally high repeatability – to allow for accurate comparison of consecutive experiments. In addition to a description of the CAC test-rig, Section 6 evaluates the performance, repeatability and adaptability of the completed test-rig.

Section 7 concludes this report with a coverage of the results gained using the BT adaptive CAC algorithm. Initially this takes the form of assessing the algorithm using threshold values calculated by our BTL collaborators although it is expected that it will be necessary to empirically establish the relationship between threshold value, CLR and mean line utilisation through a more extensive series of experiments. The procedure of establishing empirically the relationship between the threshold value, CLR and mean line utilisation is extended to two video-derived traffic sources. Each of the two video sources presented to the network different values of "offered load" and will give an opportunity for a comparison in the behaviour of a threshold-based CAC algorithm under two different load circumstances. The next stage of results involves using a mixture of traffic types to give a range of offered load into the network system. The results gained through the use of a range of offered load values enables the establishing of the effect of offered load on the relationships that exist between CLR, the thresholding value and the mean line utilisation. Section 7 closes with a comparison of the results gained using the BT adaptive CAC algorithm with results gained using two other CAC algorithms. The comparison enables the advantages and disadvantages of each CAC algorithm to be clearly contrasted.

2 Traffic Sources

Two types of sources are discussed in this section. One source type is a theoretical model source whose characteristics are both easily simulated and well understood. The second source type has been created from video stream data and represents a common real-world network application.

In order to both both make comparison with theoretical results and test predicated values for the CAC algorithm we use a theoretical source model. By using a source that can be easily simulated, such as a theoretical source model, the results gained from theoretical models of the CAC process can easily be compared with results gained from a CAC implementation that uses traffic sources with the same characteristics. In this way the theoretical results can be verified in a practical implementation.

In comparison, the traffic representing a video stream data has been created from real video data. A traffic stream is created that represents the carriage of the video data as network traffic. In this way we have a reproducible but not easily model-able traffic source representative of real-world traffic streams.

The four sources used in this study are listed, alongside their characteristic parameters, in Table 2.1.

Name	source type	PCR	SCR	MBS
TP10S1	theoretical	10Mbps	1Mbps	25
TP20S5	theoretical	20Mbps	5Mbps	50
VP10S1	video stream	10Mbps	1Mbps	25
VP5S2	video stream	5Mbps	2Mbps	50

Table 2.1: Traffic sources used in this study.

2.1 Traffic representation in ATM

During this study a well-defined form is used to represent ATM traffic. This form is needed to represent streams of traffic to be transmitted by traffic generators. If these streams are generated off-line, it involves the creation of a list of integers each representing a cell. In this way the traffic generation needs a simple play list of cells that counts the timing of cells, rather than the content of the cells or any other aspect. This comes about because ATM traffic can be represented as a stream of cells and the spaces between cells. Thus the traffic stream can be described as a series of integers, each integer counting inclusively the number of cell-slots between one cell and the next in the stream. Figure 2.1 illustrates how a stream of cells of traffic can be characterised as a series of Inter-Cell Times (ICTs).

Throughout this study, ICTs are used to represent streams of traffic. In this way streams of traffic can be indicated with a stream of integers and not the full data of the traffic stream. Such an approach reduces the complexity of producing and reproducing traffic streams as required.

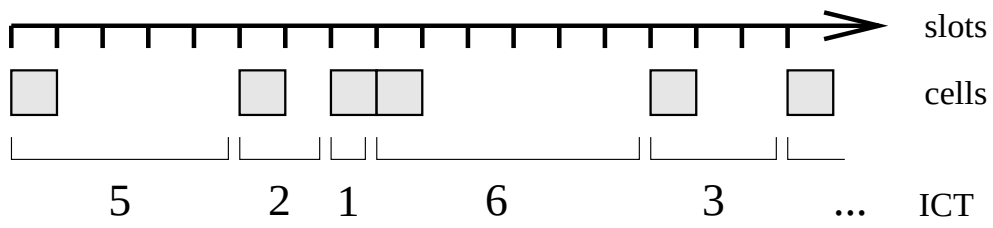


Figure 2.1: Inter-Cell Times representing a stream of traffic.

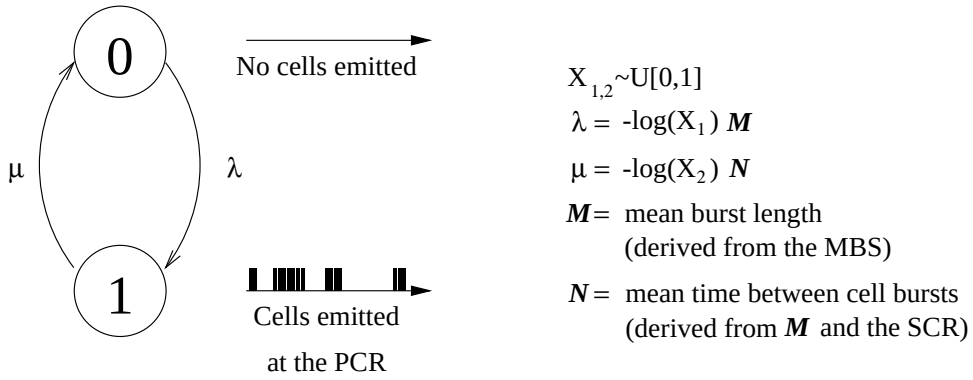


Figure 2.2: Markov 2-state on-off generator.

2.2 Sources based on Theoretical Models

The sources in this study that are based on a theoretical model are called **TP10S1** and **TP20S5**. The theoretical model used is a Markov 2-state on-off source as shown in Figure 2.2. The burst sizes and inter-burst spacings each have an exponential distribution with means derived from the MBS and SCR respectively. In the on-state, the cells of a burst are emitted at PCR.

The behaviour of traffic from the source is based around the uniformly distributed random variable X and the traffic properties of PCR, SCR and MBS. The variable X in turn, is based on a pseudo-random number generator. As a result, several traffic generators can have consistent traffic properties of PCR, SCR and MBS yet, through the use of different seeds to the pseudo-random number generator, the generators will create a stream of cells that will differ over time in the cell level characteristics of burst length, burst size and inter-burst spacing.

2.2.1 TP10S1

The TP10S1 traffic type has a PCR of 10Mbps and an SCR of 1Mbps. The cell bursts themselves have an MBS of 25 cells. This means cell bursts will be transmitted at 10Mbps,

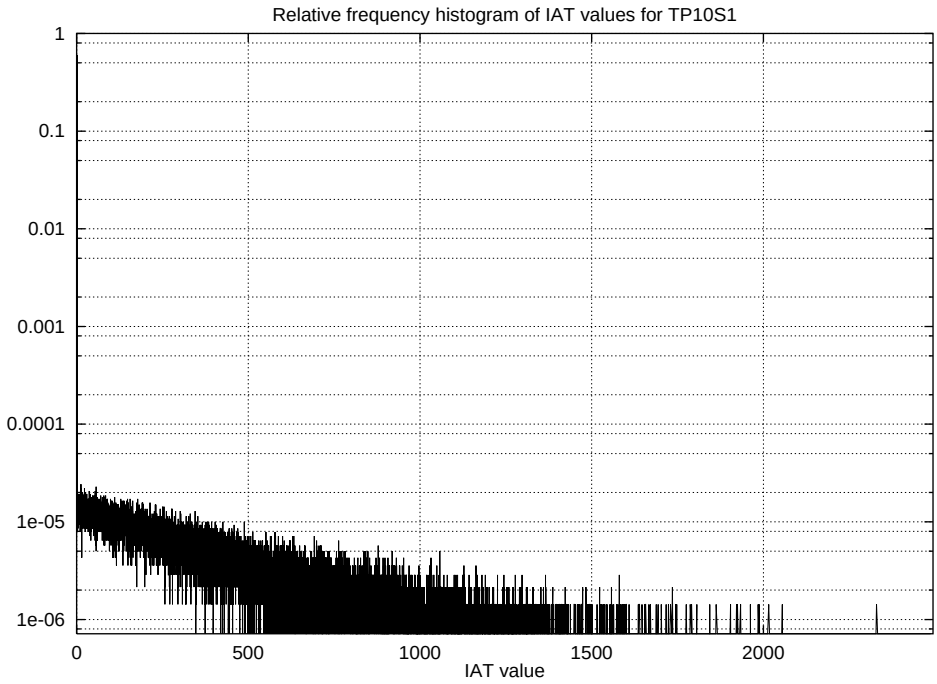


Figure 2.3: The histogram of ICTs for a single TP10S1 source.

the exponentially distributed cell bursts have a mean length of 25 cells and the exponentially distributed inter-burst spacing gives an SCR of 1Mbps.

Figure 2.3 is the histogram of inter-cell spaces for a single traffic source with TP10S1 characteristics. While unclear in the figure, there is a large peak at an ICT of 10, this corresponds to the ICT of the PCR. Such a large peak occurs because the bursts of traffic generated in this source are transmitted at the PCR.

The traffic created by the TP10S1 source is random with a mean activity equal to the SCR of 1Mbps. Figure 2.4 shows the output of one generator and Figure 2.5 shows the multiplexed output of ten independent generators.

2.2.2 TP20S5

The theoretical source TP20S5, like the source TP10S1, is based on a 2-state Markov on-off source. TP20S5 has different traffic characteristics, it has rate characteristics of a PCR of 20Mbps and SCR of 5Mbps. The cell bursts themselves have an MBS of 50 cells. This means cell bursts will be transmitted at 20Mbps, the exponentially distributed cell bursts have a mean length of 50 cells and the exponentially distributed inter-burst spacings give an SCR of 5Mbps.

However, the TP20S5 source was not used beyond the initial study. The decision not to use TP20S5 in CAC evaluations was taken because, in hindsight, the source parameters were too high in comparison to the link bandwidth and buffer length of the ATM switch in the test network.

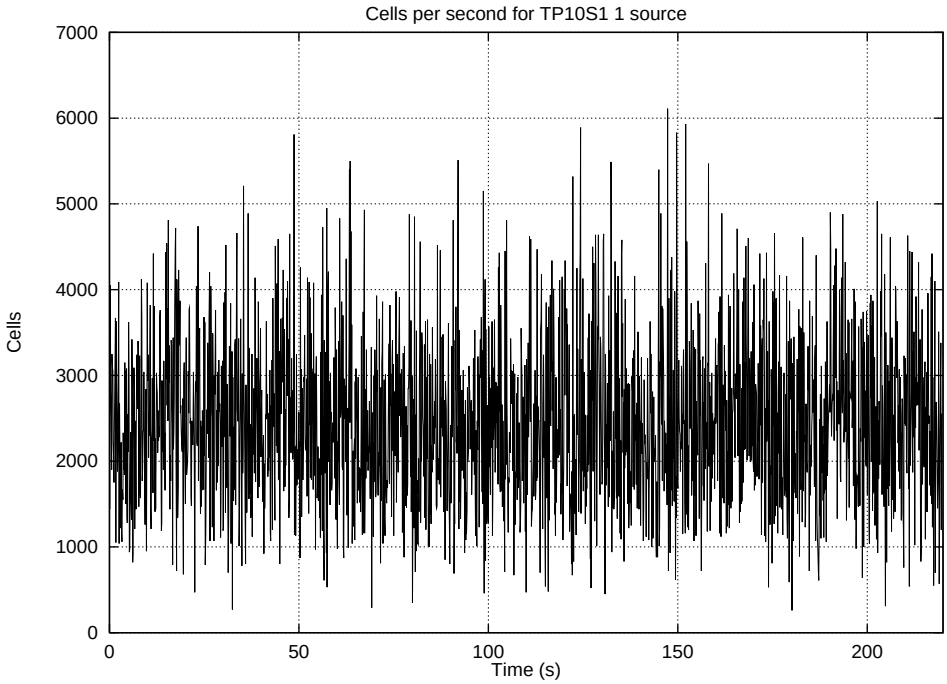


Figure 2.4: Traffic output from a single TP10S1 source.

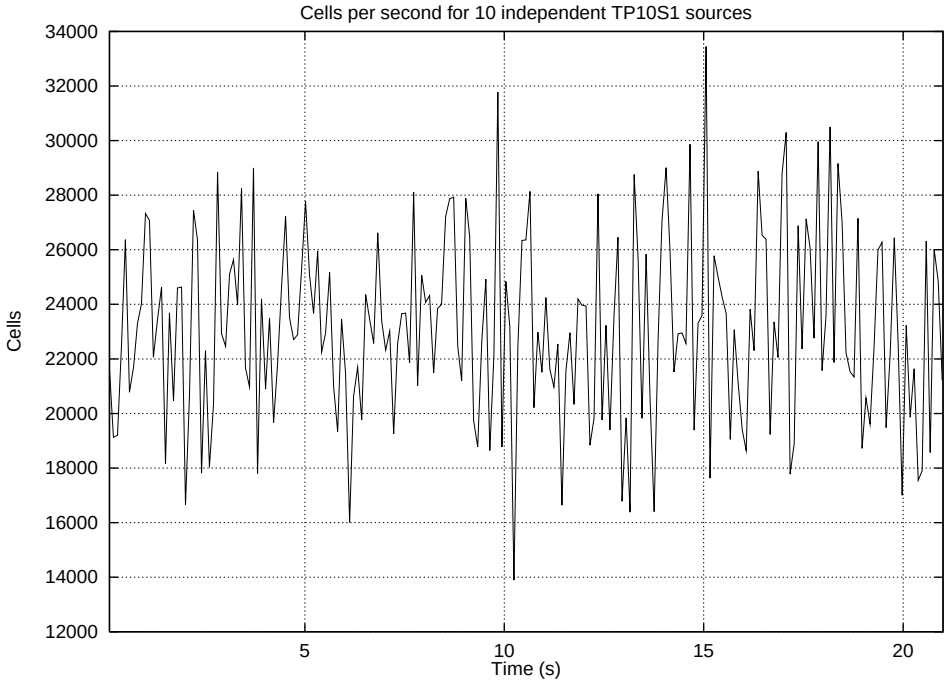


Figure 2.5: Traffic from ten independent TP10S1 sources.

2.3 Sources based on Video Streams

The sources in this study used to represent real-world, rather than theoretical, traffic are based on video data streams. These traffic sources are called **VP10S1** and **VP5S2**. Each of these sources represents the carriage of video stream data in ATM cells along a stream with pre-defined values of PCR, SCR and MBS. Through the selection of PCR, SCR, MBS we can create a traffic source whose broad characteristics are the same as our theoretical source, yet exhibit significant differences at the cell level characteristics of burst length, burst size and inter-burst spacing.

Each of the two video sources is created from a data file containing a sequence of integers. Each integer denotes the number of bytes that results from a frame of the video compressed using MPEG 1 [24].

The conversion of video data into a cell stream involves the conversion of large, periodic sets of data (a frame) into variable numbers of cells. The conversion process must allow specification of the SCR, PCR and MBS values. Such a conversion could occur in several different ways and these methods, as well as the rationalisation for the method chosen, are given in the context of a variety of methods to choose from.

The easiest method would be to convert the whole frame into a large block of ATM cells, and to then transmit these cells at the PCR for the connection. Such a method has the advantage that the PCR can be specified, additionally the desired SCR can be selected by adjusting the rate at which frames are transmitted. Figure 2.6 (a) gives a profile of the bytes per second that might occur in this encoding system. This method is not satisfactory, the main problem is that associated with cell burst sizes.

For a frame consisting of 75,000 bytes[1] a conversion of this data into ATM cells would create a single burst of 1,563 cells in length. A typical ATM switch would not carry buffers sized to cope with such a large burst, certainly in our experiment the buffer was 100 cells in length, and while the burst may be at only a percentage of the total link bandwidth, only a few connections of this type of traffic would quickly overflow the buffering capacity of a switch causing high loss in the encoded frames. Additionally, this method could not achieve a nominated mean (cell) burst size. The mean burst size would instead be directly related to the mean frame size.

An alternative approach would be to space the transmission of the cells of each burst out over the entire duration of each frame. Figure 2.6 (b) gives a profile of the bytes per second that might occur in this encoding system. This technique is not unusual; it is commonly used when matching the output of a 'bursty' device to a low bandwidth transmission path. It can be seen that while this would make the traffic more pacific, reducing the potential for buffer overrun at the switch, it does not really address the problems of the former method. Additionally, the PCR is no longer able to be specified as cells are transmitted at a rate derived from the frame rate. The overall mean of SCR can still be specified by using an appropriate value for the frame rate.

An improved method would be to divide the frame into blocks of a fixed size and

[1]The value of 75,000 bytes is not an arbitrary selection, it is the mean frame size of one of the video sequences used in this study.

transmit these blocks at the PCR. In this way the PCR can be specified, the SCR can be achieved by selecting an appropriate frame rate and the MBS can be achieved by selecting the block size as the MBS value. Figure 2.6 (c) gives a profile of the bytes per second that might occur in this encoding system. The disadvantage of this system is there is no variance in the block size, the block size is set at the value of the MBS.

A final alternative technique would be to transmit the frame using a fixed number of pieces per frame. This system of dividing a frame into pieces occurs in commercially available codecs [27]. The traffic is bundled into pieces, or more correctly 'slices', each slice corresponding to a region in the MPEG coding standard [24]. Each slice includes an AAL5 [8, 9] wrapper thus each slice can be transmitted in a manner that allows error detection. The name 'slice' is derived because each is a horizontal region of the original image. A slice technique is used in image transmission because it reduces the latency with which an image can be reconstructed; slices of the image can be reconstructed as it is received. Thus the compression and/or decompression process can occur in parallel with the transmission of an image frame.

In addition to being a common commercial technique, slices allow transmission of the blocks at PCR, the SCR can be set using an appropriate frame rate and by using an appropriate number of blocks per frame, the desired MBS can be achieved. This technique has one disadvantage not easily overcome; it has a periodicity as the bursts occur at a regular rate. Figure 2.6 (d) gives a profile of the bytes per second that might occur in this encoding system. This final method is the technique used in this study for video streams into traffic sources..

Formula can be used to calculate the required frame rate and the required number of slices per frame that need to be used to achieve the desired PCR, SCR and MBS. The formula used to convert the frame size data into slices is as follows:

$$s = F/(48 \times B - H)$$

where s is the slices per frame, F is the mean frame size, B is the mean burst size and H is the overhead per slice (such as AAL5 encoding).[2] In this way an overhead per slice is constant and the slices per frame can be determined using the mean frame size which in turn can be pre-calculated from the MBS.

In an MPEG stream, the frame rate may be a parameter that is set constant for the duration of a video stream. The desired SCR can be achieved by selecting the correct frame rate. The formula is:

$$r = (S/(F + s \times H)) \times (8 \times 48/53)$$

where r is the frame rate, S is the desired SCR, F is the mean frame size, s is the number of slices per frame and H is the overhead per slice.[3]

[2]The value 48 takes the MBS to bytes not cells, the mean frame size and the overhead per slice are also in bytes.

[3]The 8, 53 and 48 are required for conversion. The SCR is in bits per second.

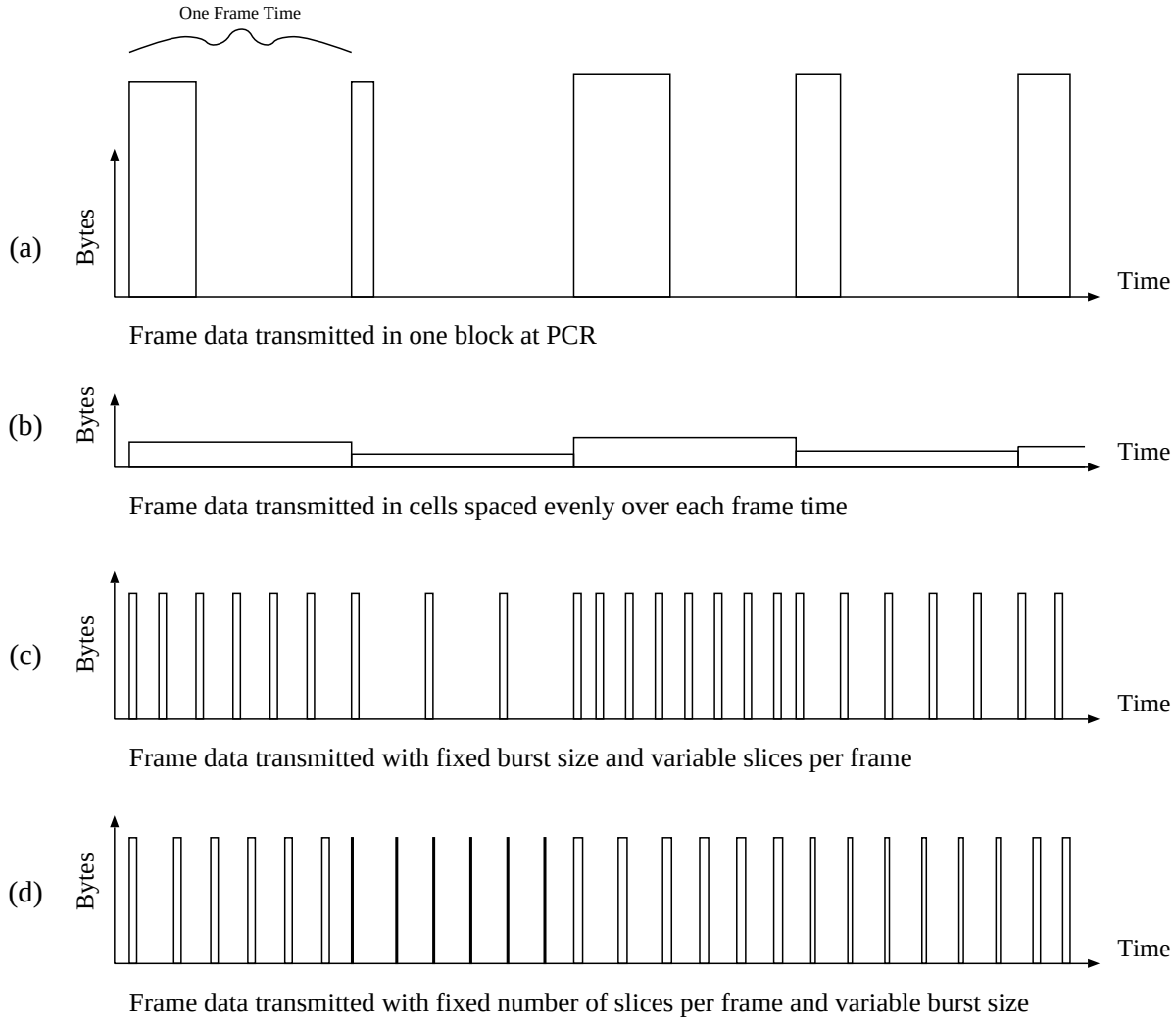


Figure 2.6: Methods for encoding frames into cells for given PCR, SCR and MBS values.

Mean	Var	Std. Dev.	95 % CI
7.516118E+04	3.786213E+08	1.945819E+04	6.609768E+02

Table 2.2: Statistical information on frame sizes of encoded "Mr Bean" (3,334 samples).

Frames per second	1.4798
Slices per frame	63.883

Table 2.3: Parameters used to convert the "Mr Bean" video stream into the VP10S1 traffic stream.

Through the use of the two formulæ above, combined with the mean size of the frame, the desired MBS and SCR values can be achieved. The PCR is used as the rate at which each burst is to be transmitted.

2.3.1 VP10S1

The traffic source VP10S1 is based on the conversion, using the method described in Section 2.3, of a list of frame sizes. The lists of frame sizes have come from a study by Rose [31].[4] Rose has converted several different sorts of video, such as feature movies, conferences and news reports into lists of frame sizes. The frame sizes are the number of bytes each frame would require once encoded using MPEG 1 encoding. The encoding of several episodes of "Mr Bean" became the source of traffic for our first video based source.

The frame sizes per frame number profile of the encoded "Mr Bean" video is shown in Figure 2.7. Statistical information about the frame sizes is shown in Table 2.2. The distribution of frame sizes is shown in Figure 2.8. Using the formula stated in Section 2.3, values can be established for the conversion of the "Mr Bean" video stream into a suitable traffic source. The VP10S1 has characteristics similar to the TP10S1 source, a PCR of 10Mbps, SCR of 1Mbps and an MBS of 25 cells.

Using the conversion parameters of Table 2.3, we obtain the VP10S1 traffic source from the "Mr Bean" video stream. The cell per second profile of VP10S1 is shown in Figure 2.9. In Figure 2.10, the histogram of cell burst sizes for VP10S1 is shown against the histogram of cell burst sizes for the TP10S1 source of Section 2.2.1. It is clear that the traffic characteristics of the VP10S1 differ greatly from the TP10S1 characterstics. Figure 2.11 shows the distribution of ICT values for the VP10S1 traffic source; Table 2.4 lists the characteristics of this source.

Mean	Var	Std. Dev.	95 % CI
9.9674E+01	3.95356E+05	6.28774E+02	1.04178E+00

Table 2.4: Statistical information on ICT values in the VP10S1 traffic source.

[4]The data are generally available at `ftp://ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG/`

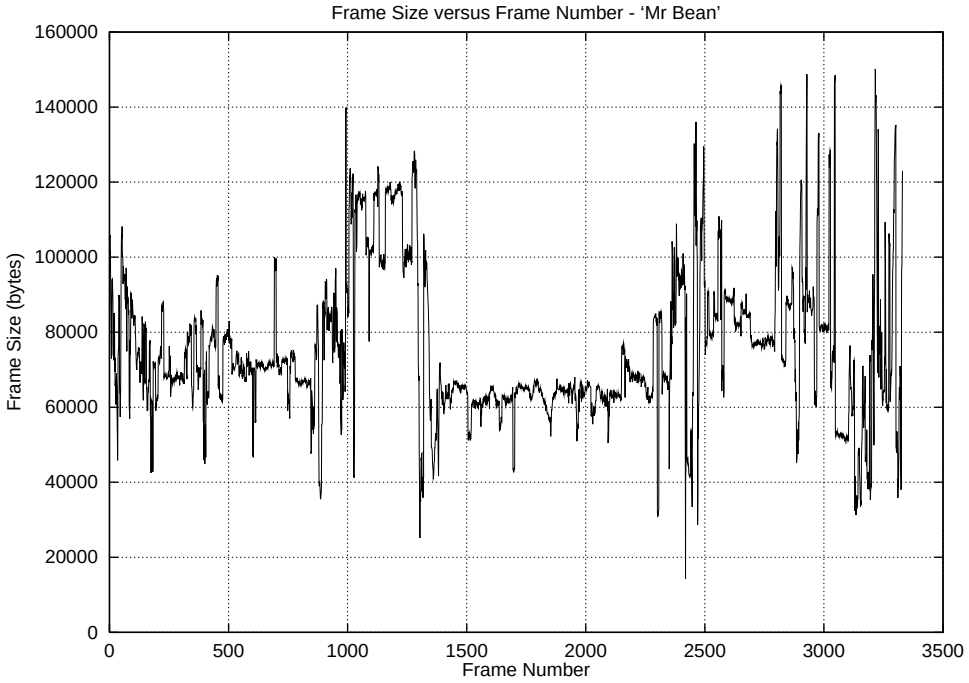


Figure 2.7: Frame sizes versus frame number for encoded "Mr Bean".

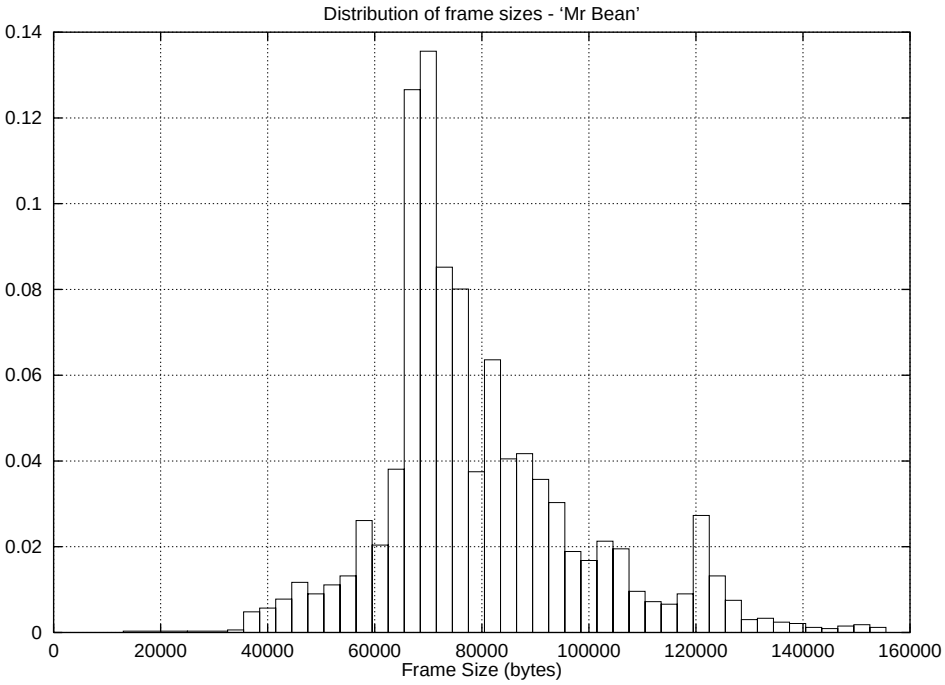


Figure 2.8: Distribution of frame sizes for encoded "Mr Bean" (3,334 samples)

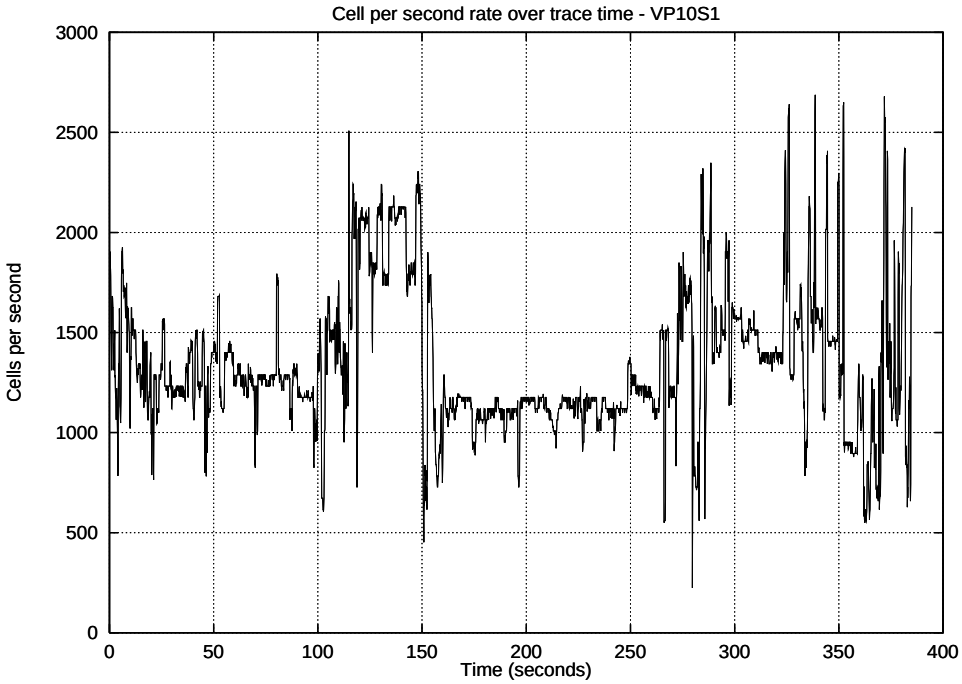


Figure 2.9: Cell per second rate versus time for the VP10S1 IAT stream.

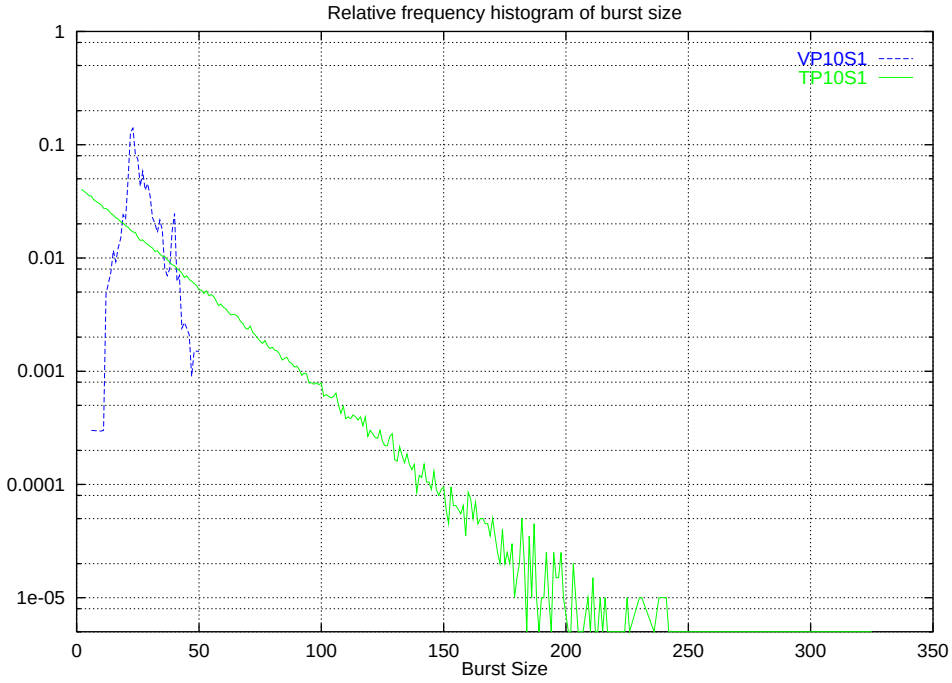


Figure 2.10: Relative frequency histogram of cell burst sizes for VP10S1 and TP10S1.

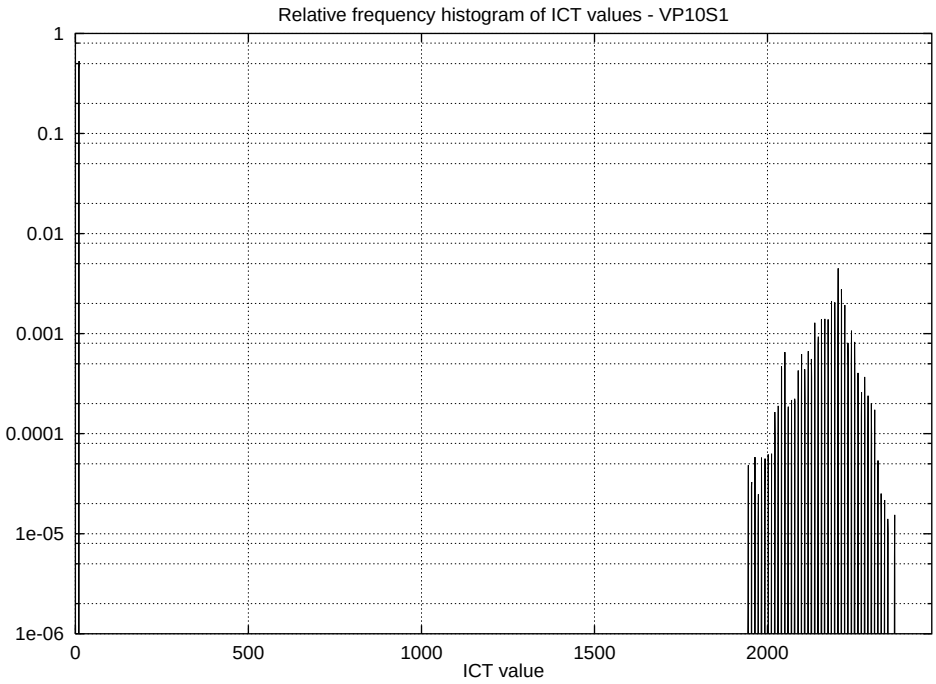


Figure 2.11: Relative frequency histogram of ICT values for VP10S1

Mean	Var	Std. Dev.	95 % CI
2.779120E+04	3.911525E+07	6.254219E+03	2.964988E+01

Table 2.5: Statistical information on frames sizes in the "Star Wars" video stream (171,000 samples).

2.3.2 VP5S2

Like the VP10S1 traffic source, the VP5S2 traffic source is based upon a list of frame sizes that would result from the encoding of a video stream using MPEG 1 image encoding. To create the VP5S2 source, the "Star Wars" video frame data of [18] was used.

The frame size per frame number profile of the encoded "Star Wars" movie is shown in Figure 2.12. Statistical information about the frame sizes are shown in Table 2.5. The distribution of frame sizes is shown in Figure 2.13. For the VP5S2 traffic source it was desirable to create a source with characteristics different from the VP10S1 source; to this end VP5S2 has a PCR of 5Mbps, SCR of 2Mbps and an MBS of 50 cells.

Using the conversion parameters of Table 2.6, we obtain the VP5S2 traffic source from the "Star Wars" video stream. The cell per second profile of VP5S2 is shown in Figure 2.15. In Figure 2.14, the distribution of cell bursts for VP5S2 is shown. Figure 2.16 shows the distribution of ICT values for the VP5S2 traffic source; Table 2.7 lists the characteristics of this source.

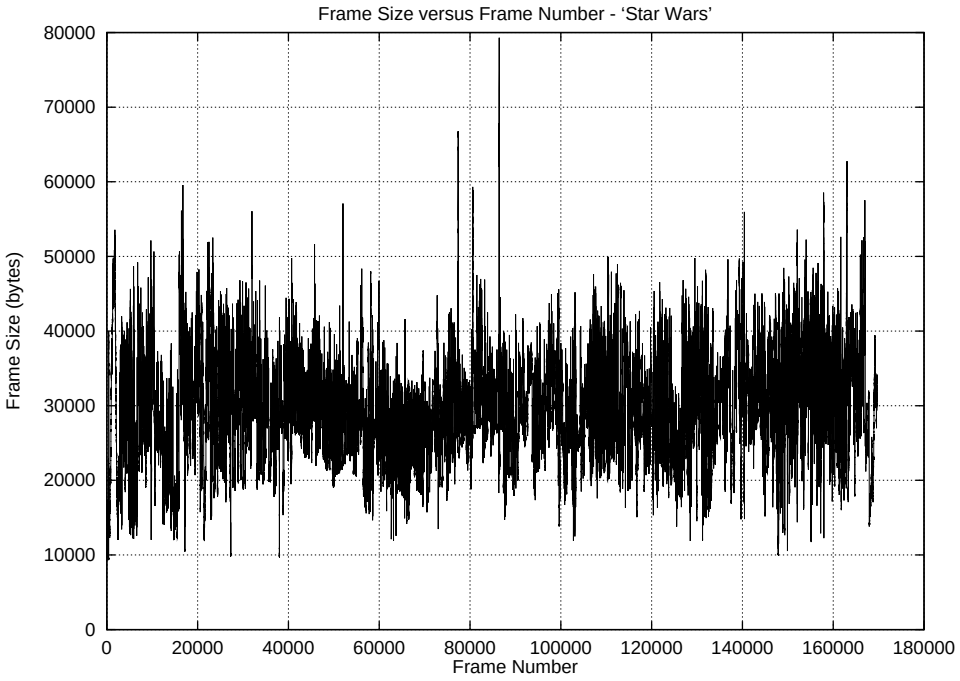


Figure 2.12: Frame sizes versus frame number for "Star Wars" video stream.

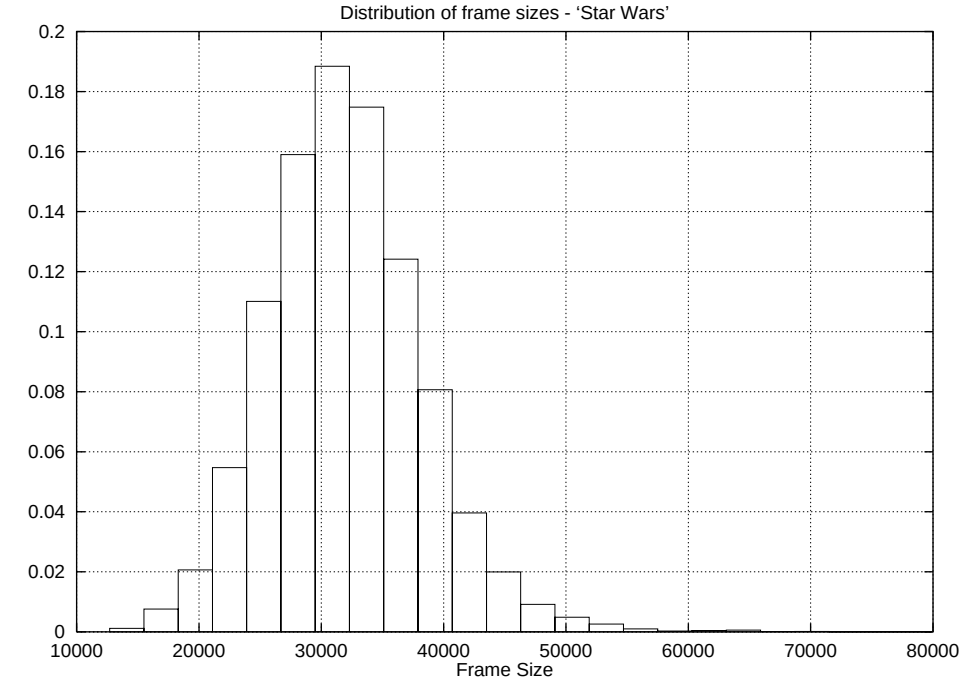


Figure 2.13: Distribution of frame sizes for "Star Wars" (171,000 samples).

Frames per second	8.278
Slices per frame	77.740

Table 2.6: Parameters used to convert the "Star Wars" video stream into the VP5S2 traffic stream.

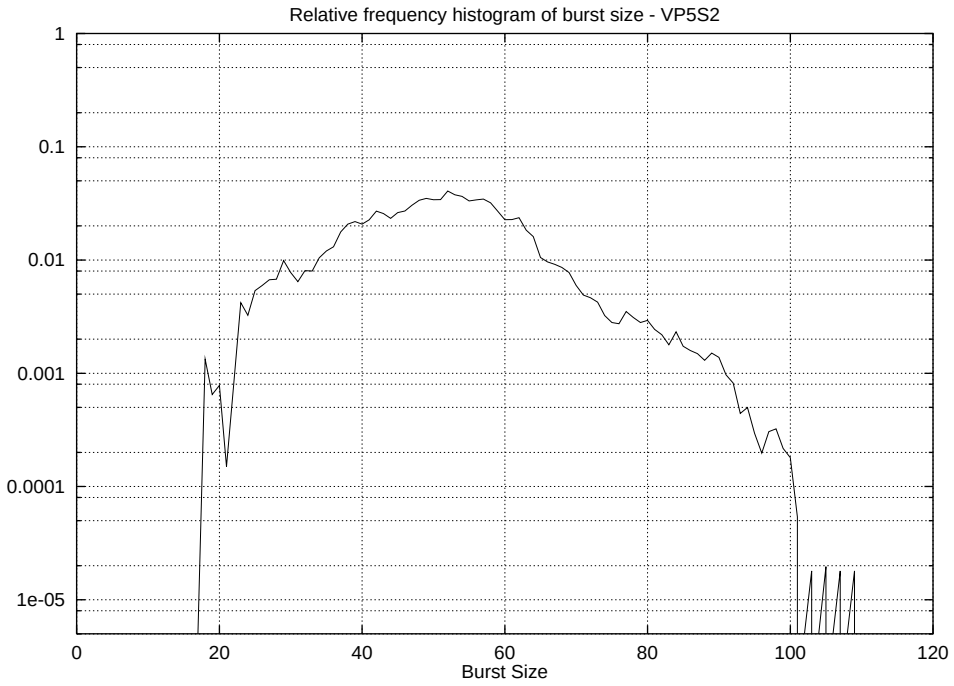


Figure 2.14: Relative frequency histogram of cell burst sizes for VP5S2.

Mean	Var	Std. Dev.	95 % CI
6.20284E+01	6.84391E+04	2.61609E+02	1.02571E-1

Table 2.7: Statistical information on ICT values of VP5S2 traffic source.

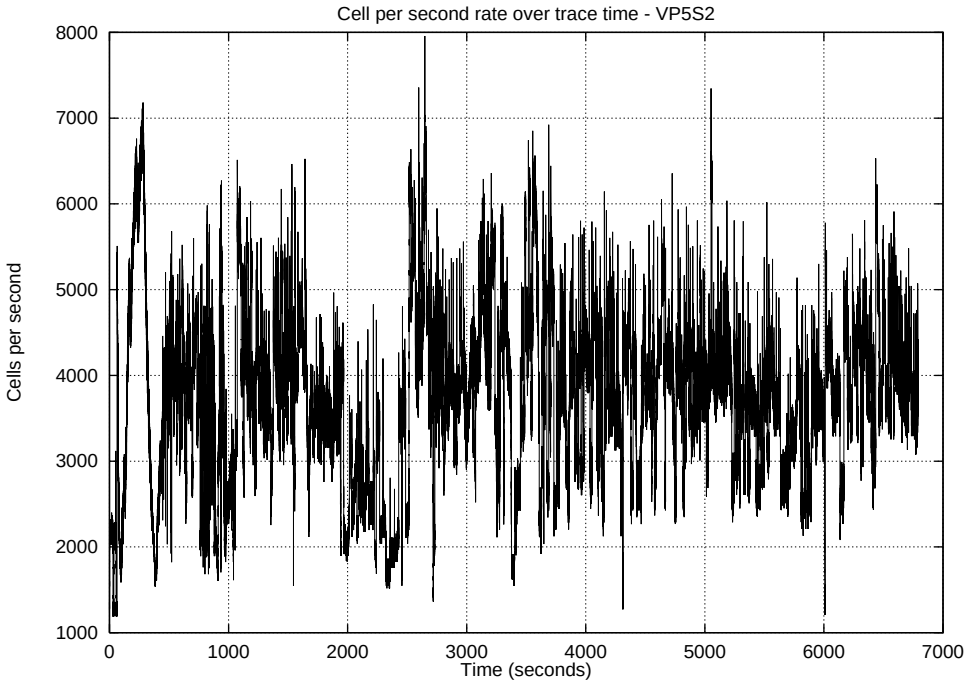


Figure 2.15: Cell per second rate versus time for the VP5S2 ICT stream.

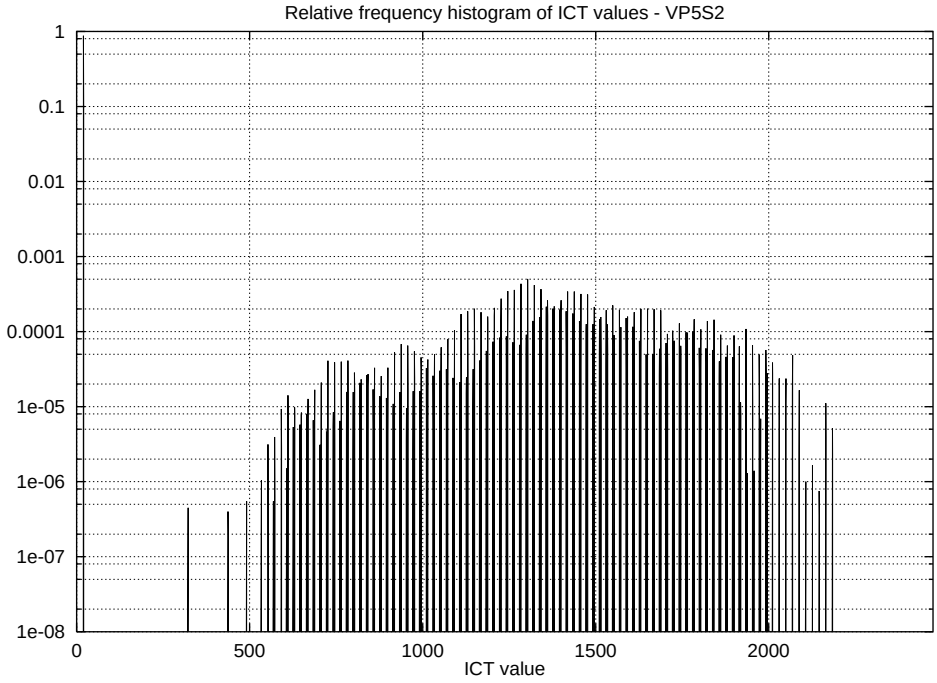


Figure 2.16: Relative frequency histogram of ICT values for VP5S2.

3 Theory

This section discusses the functioning of CAC algorithms based upon thresholding techniques and how these algorithms adapt to the varying of network conditions. This section covers the concept of "offered load" under which the network may be placed and how this concept normalises a number of differing network parameters making threshold techniques more manageable. The BT adaptive CAC algorithm, also referred to as the Key CAC algorithm, is discussed and parameters for the evaluation of the CAC algorithm are given. Finally, aspects of time scales as they relate to the functioning of the CAC algorithm are covered.

3.1 Adaptive threshold based CAC mechanism

A threshold based CAC mechanism is one that allows a new connection to be admitted into the network if the measured traffic level is equal to or below a predefined level or threshold. The threshold is calculated based on a variety of different rules and assumptions but the important factor at this stage is not how it is calculated, but rather that it is pre-calculated for use by the CAC mechanism.

The use of a threshold based CAC algorithm is shown in Figure 3.1. Connection A requests a connection into the network. The CAC makes a current bandwidth sample; the value is below the pre-calculated threshold. The CAC can admit the new connection A into the network. Now new connection B attempts to connect to the network. The CAC makes another sample of the current bandwidth; the value now is above the pre-calculated threshold. Because the value is above the pre-calculated threshold the CAC rejects the new connection B, not allowing it into the network.

For a pre-calculated value of the threshold, an arbitrary level of activity can be achieved in a network. The crucial element of threshold based CAC mechanisms is the selection of the threshold value. A threshold value is selected so that the network link will have particular characteristics. A typical set of characteristic is to balance the CLR affecting connections against the number of connections that will be refused entry to the network.

A particular threshold is calculated on the assumption of the use of one particular traffic type. A threshold based CAC algorithm can be made to adapt to different traffic types by dynamically selecting the threshold value to be used. However a new threshold would be required for each change in the traffic types active in the network.

In addition to the traffic type affecting the threshold, changes in the threshold could be required if there were changes in the length of time connections are in place (the mean call holding time or MCHT) or changes in the rate at which new connection requests are arriving (the mean connection arrival rate of MCAR). Changes in either the MCHT or MCAR will change the number of connections potentially in progress on the line; as an example if the MCHT is reduced and connections are not as long lived, there may not be as much traffic in the network and the threshold value could potentially be increased and still achieve the same line characteristics. Together the MCHT, MCAR, and the PCR and SCR of connections are used in the derivation of threshold values.

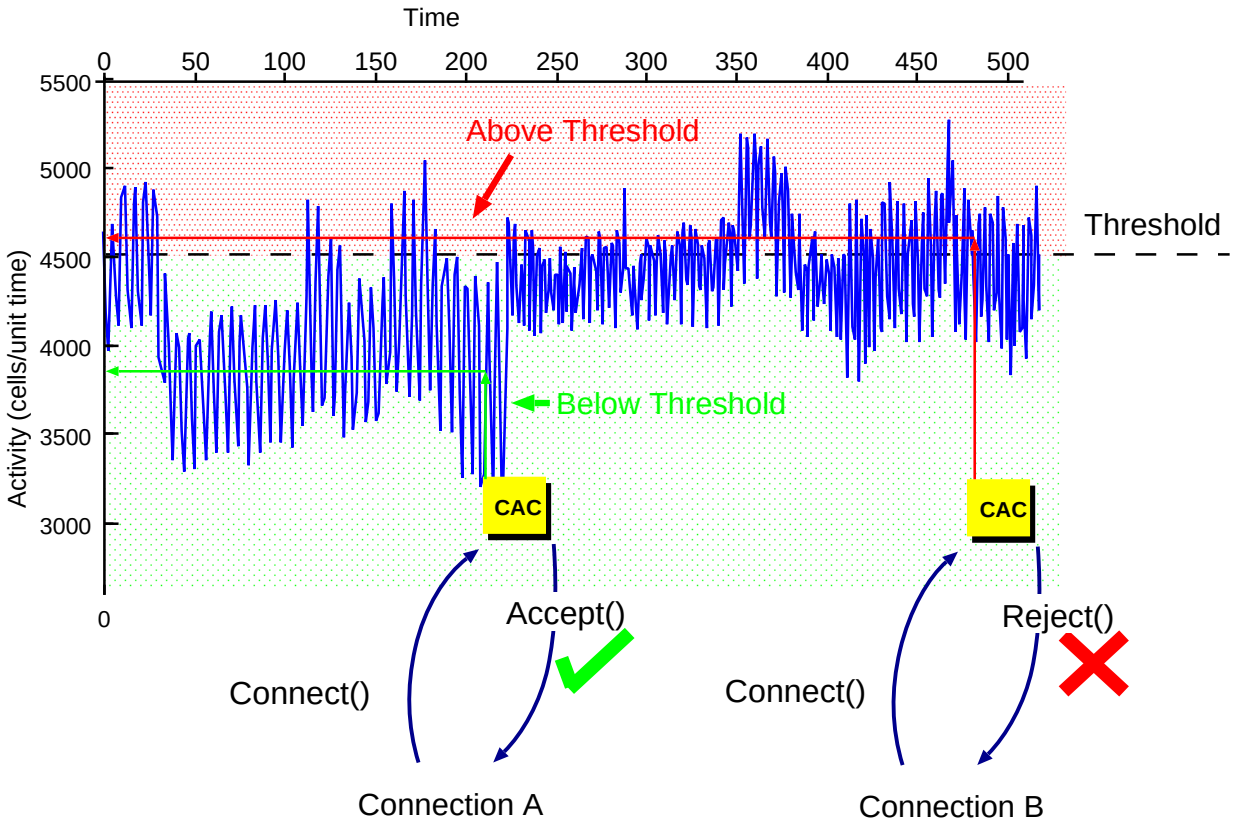


Figure 3.1: The threshold based CAC mechanism in action.

While the thresholding value can be made adaptive to changing situations in the connections on a network, it becomes clear that a new threshold value will be needed for every combination of traffic type, every combination of MCHT and MCAR, as well as any change in the network itself such as the capacity of the transmission link, the cell buffering capacity on the link or the desired CLR for the link.

3.2 Offered Load

In an effort to reduce the number of situations under which a new threshold would need to be calculated, the idea of "offered load" is applied. The offered load defines the conditions under which the CAC will be placed in a way that is consistent between different combinations of the MCAR, MCHT and connection PCR and SCR values. In this way the range of conditions for which threshold values will need to be calculated is reduced.

The offered load can be calculated from the MCAR, MCHT and an activity factor. The activity factor is an indication of the amount of transition any particular traffic source can undergo, its value is derived by dividing the connection SCR by the connection PCR. The offered load is equal to the MCAR (in connections per second) multiplied by the MCHT (in seconds per connection), in turn multiplied by the activity factor (the connection SCR divided by the connection PCR). An example could be an MCAR of 10 connections per second and an MCHT of 10s for connections with a PCR and SCR of 10Mbps and 1Mbps respectively. This would give an offered load of 10. The offered load normalises combinations of inputs, for example should the MCAR be halved and the MCHT be doubled, the value of offered load would remain constant.

3.3 Adaptation

The potential of threshold techniques can only really be exploited if the CAC can adapt the threshold value used to changes in the network conditions. With the use of offered load to reduce the complexity of calculating a given threshold, thresholds can be derived more easily as offered load will normalise out a large variety of traffic conditions. It would be easy to envisage a simple curve for deriving values for the CAC algorithm using an input of the offered load. This curve would be for a particular traffic source (or mixture of traffic sources). To create a set of useful threshold values for a variety of combinations of traffic source we could envisage a surface of threshold values. For this surface one input would be the load while another input would be the mix of traffic source types. It does not take much imagination to realise such a surface could be wildly complicated by the need to deal with many different source types.

3.4 The BT Adaptive CAC algorithm

The CAC algorithm [19, 23, 30] which is the subject of this evaluation is a threshold based system and thus makes a decision on whether to accept a connection request into the system based on whether the current line load is above a pre-calculated threshold.

Bayesian decision theory provides the framework for the calculation of thresholds which explicitly takes into account the trade-off between cell loss and connection rejection.

The algorithm is a dynamic CAC scheme using Bayesian inference and on-line measurement to estimate the user mean rate m, or its peak-to-mean ratio r. The Bayesian prior for the distribution of the source activity r is obtained from historical information on the source or its application type. Given an observation of the (instantaneous) offered load from n sources, S_n, application of Bayes theorem yields a posterior distribution on r, $f(r|s)$, which reflects the changing beliefs.

The CAC scheme accepts the next connection attempt if $S_n \leq s(n)$, where $s(n)$ is an acceptance boundary, calculated with respect to the posterior distribution $f(r|s)$, which ensures that the constraint on the expected CLR is met.

The acceptance boundaries and thus the thresholds used in this evaluation were calculated by our BTL collaborators based on connections carrying traffic with the characteristics of traffic source TP10S1 given in Table 2.1. The TP10S1 traffic source itself is discussed in detail in Section 2. The calculated acceptance boundaries are given in Table 3.1.

The acceptance boundary is calculated without reference to the transmission link rate of a network; this allows calculation independent of any particular speed network. The acceptance boundary must be converted into a value applicable to a specific network. The acceptance boundary shares a relationship with the threshold to be used by the CAC algorithm based upon a factor called the line capacity. The line capacity is the maximum number of connections that can be accepted if each of these connections were operating at their nominated PCR. Thus the line capacity is the transmission link rate divided by the PCR of the connections' traffic. For this evaluation the transmission link rate is 100Mbps and the PCR of connections is 10Mbps; therefore the line capacity is 10. To convert the acceptance boundary into the threshold value, the acceptance boundary is multiplied by the line capacity. The threshold values to be used are given alongside the acceptance boundary values in Table 3.1.

It was decided that the actual values used in this study should be limited to those achieving a CLR of 10^{-3}. This decision, taken under advice from our BTL collaborators, was on the basis that attempting to obtain lower CLR targets would give no useful data as experimental results, for CLR in particular, were reduced to insignificance when compared with the possible error range.

3.5 Time scales

The BT adaptive CAC algorithm uses the instantaneous measurement of traffic utilisation as part of the Connection Admission decision. Such instantaneous measurements are taken over a nominated period of time. The time over which the measurement is taken and the time-scale of traffic characteristics to be measured share a significant relationship in a CAC algorithm that depends on the instantaneous measurement of traffic utilisation. A notion of time-scales as applied to network traffic within a connection was first applied by Hui [22]. He noted that there were:

Offered Load	10	4	3
MCAR (per second)	10	4	3
MCHT (second)	10	10	10
To achieve a CLR of 10^{-3}			
Acceptance Boundary	4.26	5.74	7.46
Threshold (Mbps)	42.6	57.4	74.6
To achieve a CLR of 10^{-4}			
Acceptance Boundary	3.26	4.04	4.74
Threshold (Mbps)	32.6	40.4	47.4
To achieve a CLR of 10^{-5}			
Acceptance Boundary	2.59	3.08	3.49
Threshold (Mbps)	25.9	30.8	34.9

Table 3.1: Acceptance boundaries supplied by BT for evaluation.

long or seconds, tens of second time scales – such as the duration of a connection which could be seconds or minutes in length.

medium or millisecond time scales – a connection can consist of many bursts of traffic activity surrounded by periods of inactivity. This is the burst scale.

short or microsecond time scale – each burst will consist of a number of cells sent at the line rate with intervening inter-connection gaps. This is the cell scale.

The relationship between time scales and the period over which an instantaneous measurement of traffic should be taken is explored at length by Gibbens et al. [19]; it can be summarised as follows:

In order to maintain a given CLR value, any given CAC algorithm must take into account the smallest of the time-scales and the relationship of the time between cells with the size of the buffer.

In addition to the smallest time-scale, a CAC algorithm must measure over a period of time of a length that will include traffic samples from (on average) all connections currently in the network. The utilisation period, therefore, should not be greater than the MCHT, otherwise account will be made of calls no longer in the system.

This means the resolution of such an instantaneous measurement is that of a single cell while the period over which such an instantaneous measure should be measured is given as:

$$b \times C_t < t_m < H$$

Where b is the size of the buffer, C_t is the transmission time of a single cell, H is the MCHT and t_m is the time over which the measurement is to be made.

For example, should the buffer be 100 cells in length, the value of C_t be 4.4 microseconds and the MCHT be 10 seconds – the time over which the measurement is to be made, t_m,

has the following bound:

$$440\mu s < t_m < 10s$$

An implicit assumption throughout the theory of this algorithm is that instantaneous measurements of traffic can be made for each and every new connection request on which the CAC algorithm must decide. The above equations show that the period of an instantaneous measurement is not zero and the range of values include the time during which new connections will attempt to be introduced into the system. This means that there could be many measurements taken simultaneously while new connections are being decided on by the CAC algorithm.

The hardware of any existing switch is not able to perform such simultaneous instantaneous measurements with the desired resolution over the desired period. However, periodic measurements of this resolution and measurement period can be realised. The issues this introduces are discussed further in the implementation discussion of Section 7.1.

4 Initial Experiment Configuration

The experimental configuration used to conduct an initial set of experiments is described in this section. The initial experiments were designed to establish parameters of the CAC mechanism as well as parameters of the evaluation experiments to be conducted. To this end the initial experiments consisted of simple configurations of equipment to source and sink traffic sources with the instrumented ATM switch (on which most measurements would be taken) between.

A discussion of the unique characteristics of the ATM switch and a comparison of this switch with modern commercial switches is contained in this section. The physical generation of the ATM cells that make up a traffic sources is considered. The method of measuring is noted next along with the equipment used to sink traffic sources is considered. The generation of traffic sources was done using a combination of the off-line generation of sources of traffic combined with the off-line multiplexing of these sources of traffic. The methods involved and how this relates to the sourcing of cells in the network is the final part of this section.

4.1 ATM network – Fairisle

The ATM switch used is the Fairisle ATM switch developed at Cambridge. The Fairisle project developed an ATM platform to permit research into performance, architecture and management issues in ATM networks [26, 6]. It is based on a network architecture of ATM switches which are interconnected to form a general-topology ATM LAN. The Fairisle switch is based on a simple, 4×4 input-buffered crossbar. Switches of a degree greater than 4 can be constructed by using multiple 4×4 crossbars in a 2-stage delta interconnection network.

The hardware design of the Fairisle network components permits a high degree of experimental flexibility owing to the use of field programmable gate arrays. These have a number of features to support experimental work, including high-resolution clocks and hardware time-stamping of cells.

Each port is equipped with 2K cell buffers, 4 MB of RAM and an ARM RISC processor, which implements all policy-specific traffic control functions, using a flexible control plane which exports RPC-style interfaces [5, 33] for signalling and management. This enabled the implementation of a wide range of measurement facilities to accurately monitor queue behaviour at the cell level resolution.

Many manufacturers of commercial ATM equipment base their switch designs on an effectively non-blocking (output-buffered) fabric. Although this increases the complexity and cost of the switch, output buffered designs are appealing because they remove the effects of contention on the switch fabric. To construct an output buffered switch, the switching stage must be made effectively non-blocking. This was accomplished in these experiments by using two input-buffered Fairisle switches connected in tandem to implement a single output-buffered switch. The tandem pairing of switches A and B shown in Figure 4.1 are such an example.

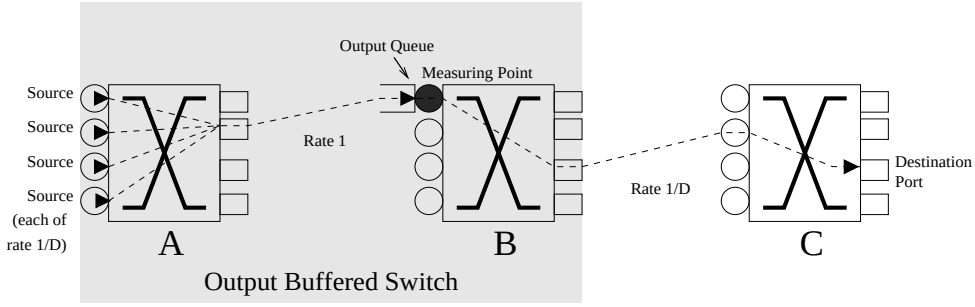


Figure 4.1: Switch topology for single stage, output-buffered switch.

In Figure 4.1 each link is labelled with its transmission-rate relative to the full line rate of 1. The transmission rates of traffic sources are scaled by a factor of $1/D$. The transmission-link-rate for the output port of the first switch (A) is set to the full link rate. Traffic arriving at the inputs of this switch is switched by the first fabric and multiplexed onto the full rate link. The output link of the second switch (B) is scaled by $1/D$. Traffic arriving at the input of this second switch is blocked by back-pressure across the fabric from the output whose link runs at the reduced rate. Queueing occurs at this point, which can be viewed as the output buffer of the switch so constructed. The output buffered switch has a switching capacity of D times the input and output transmission rates, and is therefore effectively non-blocking.

The potential offered by each port in Fairisle having a separate processor and memory means that each port can operate independently and can even perform tasks not directly involved in the routing of ATM cells on the switch. This situation means that in addition to ports being used as part of the switch itself, ports are used to perform the task of traffic sources and of traffic sink. This gives excellent flexibility in the construction of the test environment while maintaining consistency in component behaviour and interconnectivity.

4.1.1 Traffic source

When a Fairisle ATM port is operating as a source of traffic a traffic source, represented as a sequential list of ICT values, is loaded into memory prior to transmission. On command the traffic source will transmit cells with a spacing determined by the list of ICT values in memory. The path the cells will take is predetermined at configuration time. The contents of the cells is irrelevant; each cell usually contains random contents. By not having to fill each cell the port can be programmed to give maximum effort to the timing aspects, ensuring the cells are emitted exactly as per the ICT list.

Figure 4.2 shows a schematic of the traffic generator. The traffic generator sequentially scans a list of ICT values generating cells with the spacing dictated by the ICT list. A port transmitting cells can transmit a number of nominated cells, or can transmit cells for an indefinite period. The amount of memory on the port acting as a traffic source allows for the source to be loaded with 1.4 million sequential ICT values. If the port transmits more than 1.4 million cells, the ICT list is repeated. For example to transmit 10 million cells, the

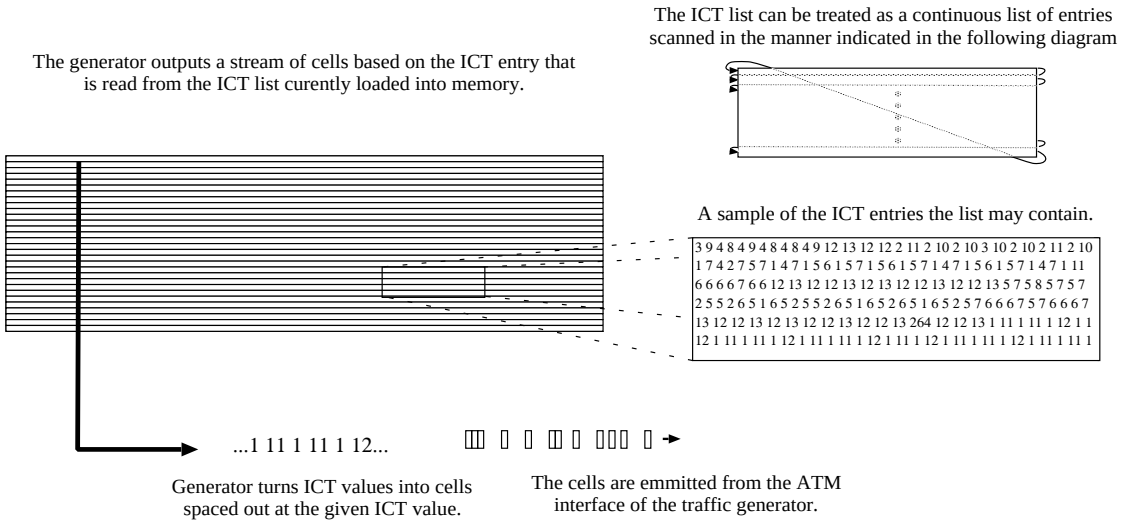


Figure 4.2: Fairisle ATM port as a traffic generator.

ICT list would be looped through over 7 times. There is no mechanism to dynamically load the ICT list as it is played. The ICT list used by the traffic source is generated "off-line", prior to use by the traffic source.

4.1.2 Traffic measurement

Fairisle has the high resolution clocks and computer resources per port to enable measurement code to be executed on the arrival of each cell; this allows the easy extraction of queue lengths, cell spacing and such values. The result is that a wide range of measurements can be taken related to the transition of cells through the ATM switch.

The most commonly used measurement in the switch was measuring the length of the output queue at any stage. The actual results were a histogram built up as a count of the length of the output queue as each cell arrived into the queue. Such a set of results can be processed to give the probability of a cell entering the queue when it is of a certain length – this was considered very useful in this early stage of the results enabling prediction of CLR for given traffic combinations.

4.1.3 Traffic sink

The Fairisle port acting as traffic sink had no significant task to perform once the experimental kit was operational. However during set-up the sink port played many useful roles. The sink port could be programmed to collect a trace of cells, recording the ICT between cells as they arrived at the port. Such a list was invaluable in ensuring the correct behaviour of the generator. The sink and source ports could be directly connected together and the output of the traffic source port could be checked against the ICT list that was

collected at the sink port. Additionally the sink port was useful in determining if cells were passing correctly through the full test environment.

4.2 Off-line generation of traffic sources

During the course of experiments it was required that the output of up to 100 traffic sources transmit cells into the network. For the simple Fairisle system this would require 100 ports alone to act as traffic sources, this number does not include the ports required to provide interconnection or the ports required to build the output buffered switch. Such large numbers of ports were not available; the solution was to have one physical port transmit the cells of more than one traffic source.

In order for one physical port to transmit the cells of more than one traffic source, the ICT to be used by the physical port must represent cells transmitted by the multiple traffic sources. To this end a software system was built that could generate one ICT list of a multiplex of cells from multiple independent traffic sources.

The off-line process by which the traces were created is outlined in Figure 4.3. The 2-state Markov generators shown in this figure are identical to the 2-state Markov generator of the TP10S1 traffic source of Section 2.2.1 (shown in Figure 2.2). Each Markov generator uses an independent random number generator in the creation of events representing cell stream. The events that each 2-state Markov generator creates are merged together and cells are queued as each event occurs. A task, operating at the period of the cell transmission at line rate, removes cells from the queue (in FIFO fashion) calculating the ICT between that cell and the previous to be removed from the queue. As the cell is removed from the queue, the software outputs this ICT value.

It is important to note that because each generator is independent – the resulting multiplex of traffic is the same as that generated through the physical multiplexing of cells created from physically independent 2-state Markov generators transmitting cells into a single queue. The queue in the multiplexing system is marked **Q**. Unlike a physical queue, this artificial queue is not bounded in length.

4.2.1 Simulated curves

Throughout the results of the initial experiment phase a simulated curve is usually plotted on curves showing the cumulative queue length. The values used for these curves do not come from a theoretical model as such but from counting the length of the queue in the generator marked **Q** in Figure 4.3. In this way the program producing the off-line ICT list is also able to generate a simulation of the behaviour of the queue that will occur in the ATM switch using the ICT list that it also creates.

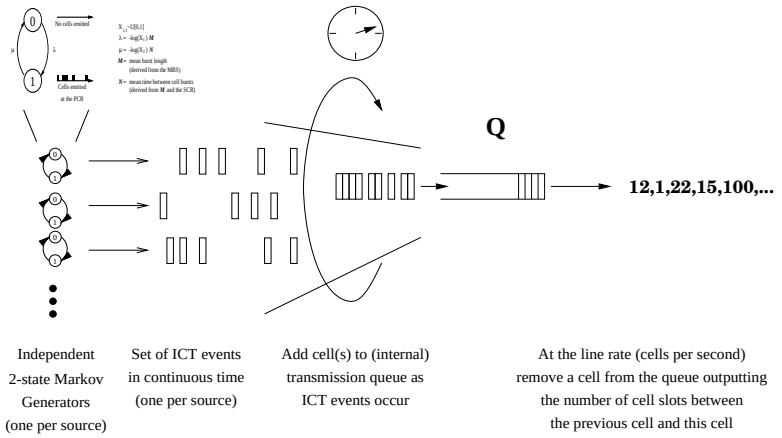


Figure 4.3: Method by which off-line ICT lists are generated.

5 Initial Experimental Results

The objective of the initial experiments was to give results to our BTL collaborators to enable the checking of effective bandwidth calculations and to investigate the relationship between CLR and buffer overflow. Additionally the initial experiments allowed the level of accuracy required for each experiment to be established; this established directly the minimum number of cells to give a balance between the run time of experiments and obtaining statistically significant values for CLR. To achieve an experiment with a satisfactory running time but with a statistically significant number of cells, experiments with 1×10^8 cells transmitted through the run was consider to be satisfactory.

The initial experiment set also measured the relationship between traffic source independence and the complementary queue length distribution (Complementary CDF). The initial results showed that the method of multiplexing traffic sources did not achieve enough independence for traffic sources and would cause serious artifacts in the experimental outcome. The direct conclusion of this problem was that a new form of traffic generator would be required for the full CAC evaluation.

The final major outcome from the initial experimental results was to not perform any further work with the theoretical traffic source TP20S5 described in Section 2.2.2. This decision, by our BTL collaborators, was made on the basis that the TP20S5 traffic source source parameters were too high in comparison to the link bandwidth and buffer length of the ATM switch in the test network, to produce any useful results.

5.1 Initial results

The initial phase of this project involved the running of more than 100 experiments. This number does not include numerous calibration experiments or those experiments run to show that testing with periodic sources would produce no useful results.

5.1.1 Method

The configuration described in Section 4 is used for the initial experiments. This configuration comprises of a Fairisle port generating cells from an ICT play list and sending this stream of cells into further Fairisle ports that comprise an ATM switch. The Fairisle ATM switch records information about the length of the output queue at the arrival of each cell for that queue.

As discussed in Section 4.2, one physical Fairisle port could potentially be sourcing the cells for many independent sources of traffic. The majority of experiments run were conducted with four physical Fairisle ports sourcing the cells of up to 92 independent traffic sources. As discussed in Section 4.2 one physical port transmits cells as per an ICT list that has been created by multiplexing the output of several (whatever the required number) theoretical traffic sources together. Most of the remaining experiments were run with two physical generators representing the output of up to 94 independent theoretical traffic sources. In each of these cases the output of the physical generator is physically

Traffic source type	Number of sources	Figure showing Complementary CDF
TP10S1	30 – 92	5.1(a)
TP10S1	88 – 92	5.1(b)
TP10S1	80	5.2(a)
TP10S1	90	5.2(b)
TP10S1	92	5.3
TP20S5	6 – 18	5.4
TP20S5	15	5.5(a)
TP20S5	18	5.5(b)

Table 5.1: Initial experiment parameters and figure numbers graphing the resulting Complementary CDF relation.

multiplexed together into the switch to give the full desired cell load at the switch. The effect on the experiments by the number of physical Fairisle ports used is discussed at length in Section 5.2.

The method used during the initial experiments was to send traffic from a number of traffic sources simultaneously into the ATM switch and extract the resulting queue length information. This information about the length of the output queue at the arrival of each cell for that queue can then be turned into graphs showing the Complementary CDF relationship. This process was repeated for different numbers of traffic sources and the whole procedure of using different numbers of traffic sources was employed for both the TP10S1 and the TP20S5 traffic source types. Table 5.1 lists the experiments' parameters together with the figure that graphs the resulting Complementary CDF relation. Each experiment ran for approximately 2 hours and each experiment involved the transmission of 1×10^8 cells unless stated otherwise.

Each of the sets of results generated were sent back to our BTL collaborators for interpretation in order to check the effective bandwidth calculations and to investigate the relationship between CLR and buffer overflow. As a result, there is only a minimal interpretation of results presented here. It was felt to be inappropriate to inexpertly interpret results that were intended for full interpretation by our BTL collaborators.

Each of the figures in Table 5.1 has a characteristic shape where the $P(Q \geq q)$ has an exponentially decaying relationship with the observed queue length up until a point at which the relationship becomes more erratic, leading to a sharp drop in the $P(Q \geq q)$. The set of samples in each experiment contains 1×10^8 cells, a $P(Q \geq q)$ of 1×10^{-8} represents the observed queue length of only one cell. Due to the low number of cell counts in the tail of each figure, the accuracy of the tail end $P(Q \geq q)$ values can be called into question.

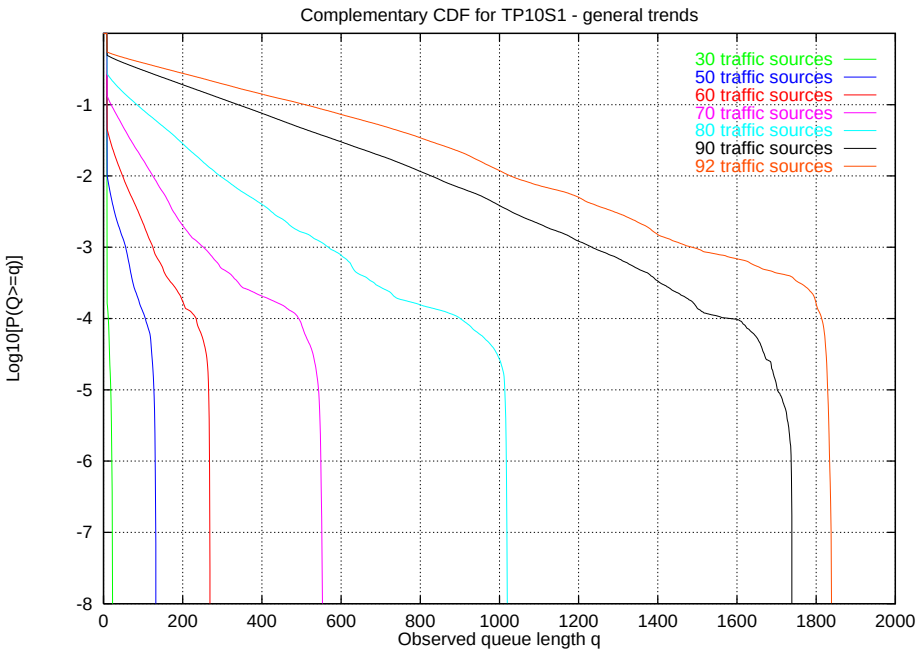
The graphs shown in Figures 5.1(a), 5.1(b) and 5.4 each depict the trend when the number of traffic generators is increased. Each graph clearly shows that as the number of sources is increased, the $P(Q \geq q)$ is increased for any particular queue length; this implies that for an increasing number of sources, a cell entering the queue will have an increasing

probability of encountering a larger queue.

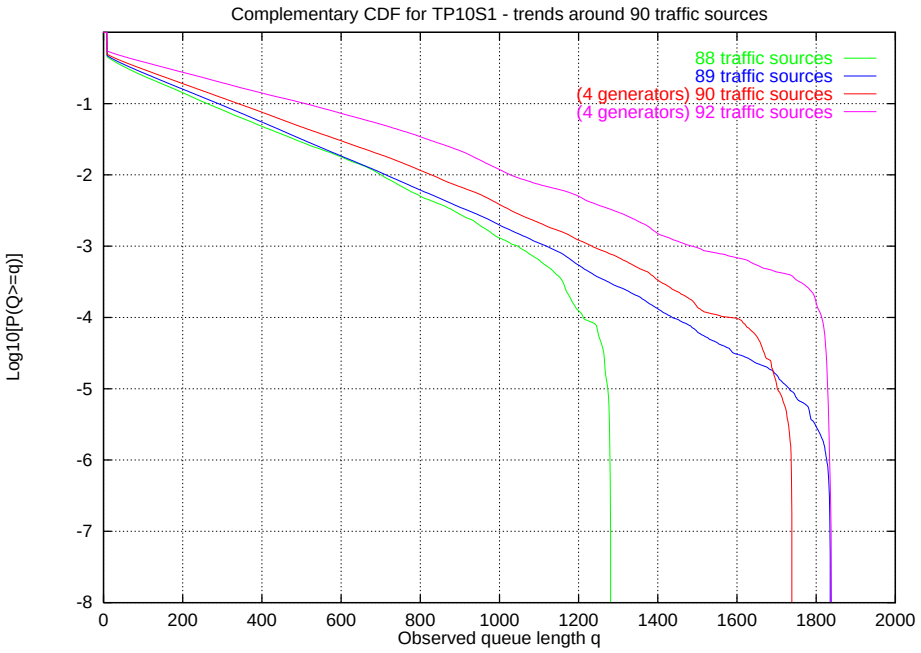
Figures 5.2(a), 5.2(b) and 5.3 show that even for the same parameters there is considerable variation in the resulting Complementary CDF. Each of these graphs has been created using a constant number of theoretical traffic sources, although the number of physical Fairisle ports used to generate the traffic of each experiment has been changed as noted. It should be remembered that each physical Fairisle port is transmitting the multiplex of numerous theoretical traffic sources. It should further be emphasised that, as discussed in Section 4.2, the multiplex consists of the traffic of a number of traffic sources, each independent of the other. The multiplexing of the traffic sources is the multiplexing of the output of the independent traffic sources, this multiplex of traffic is then transmitted by the Fairisle port.

In addition to the Complementary CDF created using the TP10S1 traffic source, Figure 5.4 as well as Figures 5.5(a) and 5.5(b) show the Complementary CDF resulting from experiments using the TP20S5 traffic source. The characteristic shape of these figures are very similar to that of the TP10S1 experiments; from this we can conclude that this Complementary CDF relationship will form independently of the PCR, SCR and MBS values for the 2-state Markov on-off traffic source.

It is noteworthy that in both Figures 5.5(a) and 5.5(b) the simulated Complementary CDF presents a larger observed queue length than actually occurs on the switch. This trend predicts that cells have a probability of encountering a smaller queue of any given length in the actual experiment compared with the situation that the simulated Complementary CDF predicts. This trend also occurs in the experiments made using the TP10S1 source.

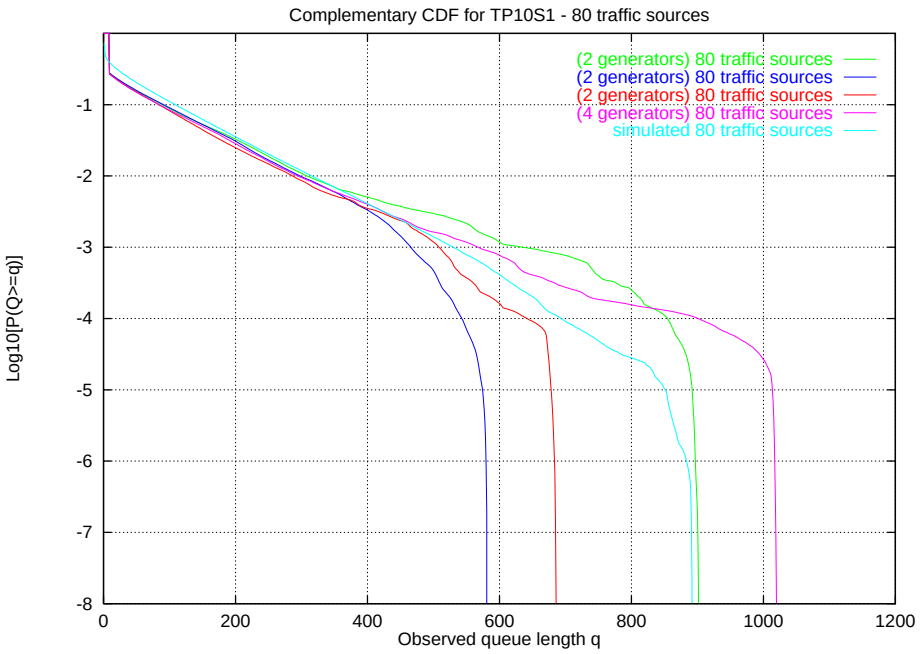


(a) General experiment trend

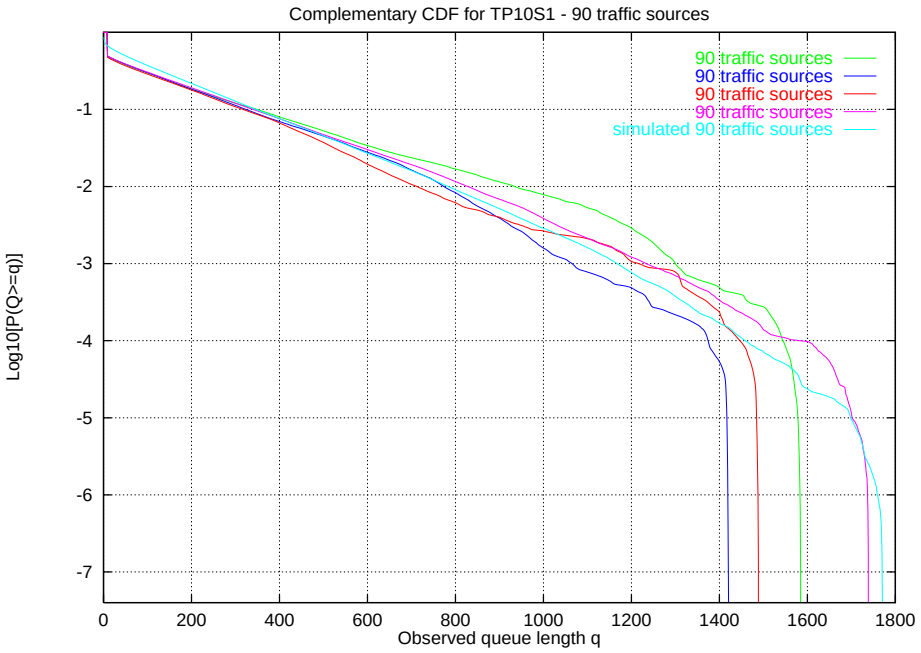


(b) Trends smaller range

Figure 5.1: Trends in Complementary CDF of experiments using the TP10S1 traffic source.



(a) 80 sources



(b) 90 sources

Figure 5.2: Complementary CDF of experiments using the TP10S1 traffic source.

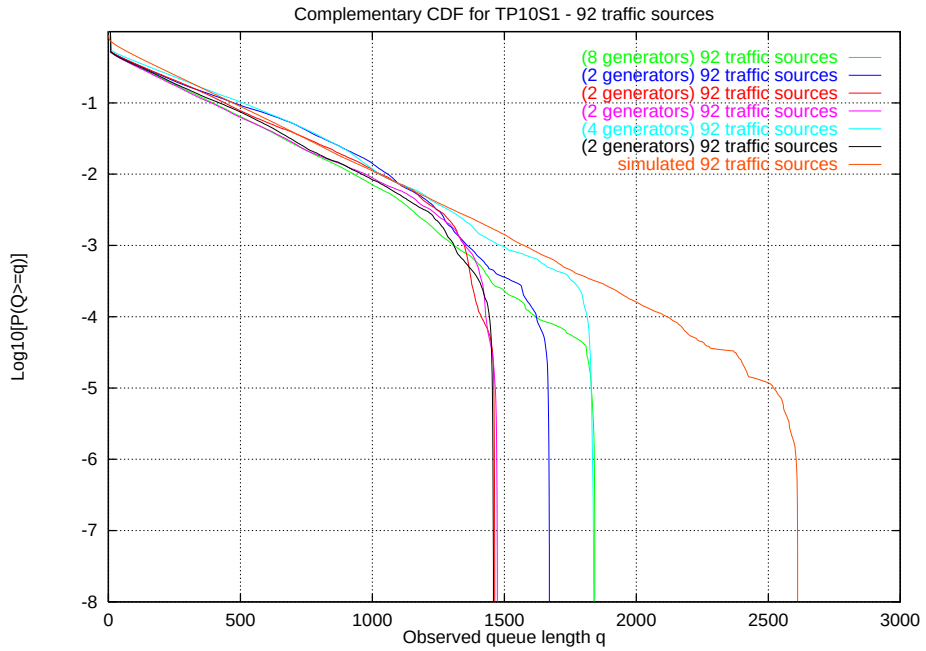


Figure 5.3: Complementary CDF of experiments using 92 TP10S1 traffic sources.

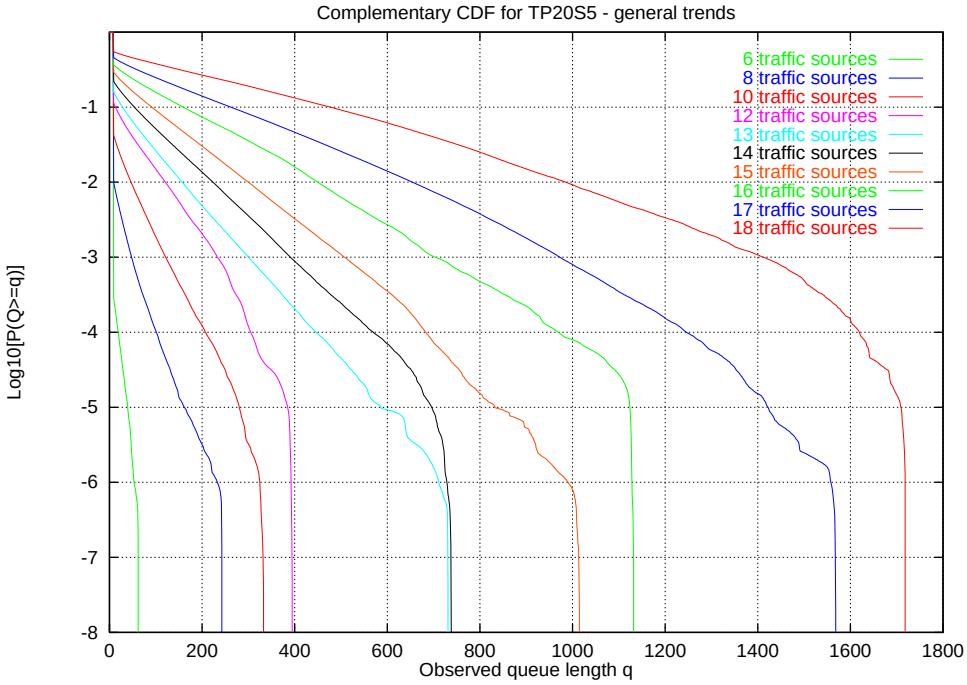
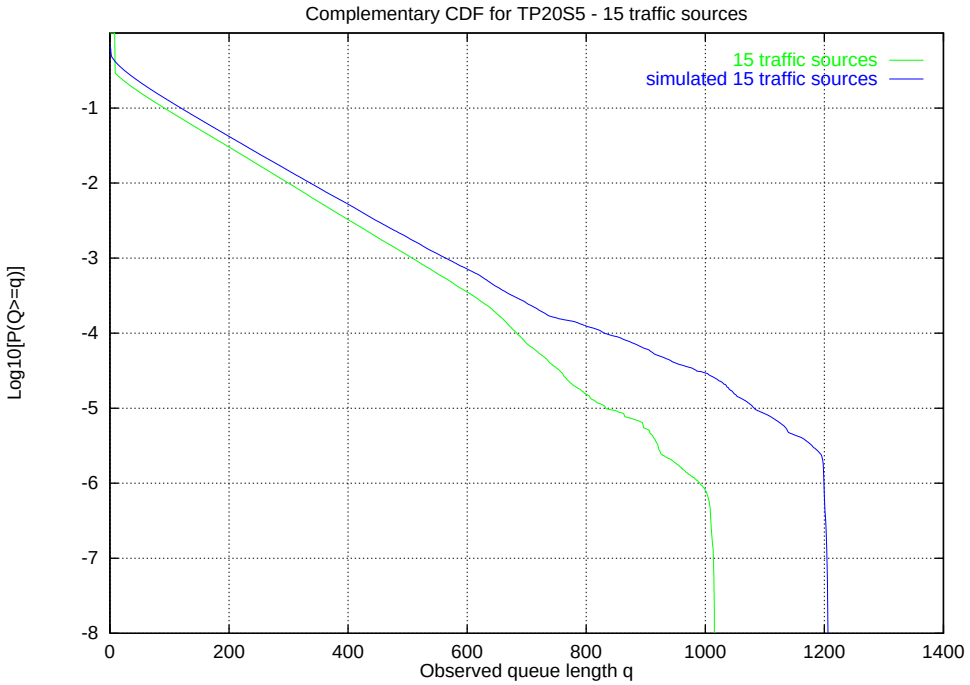
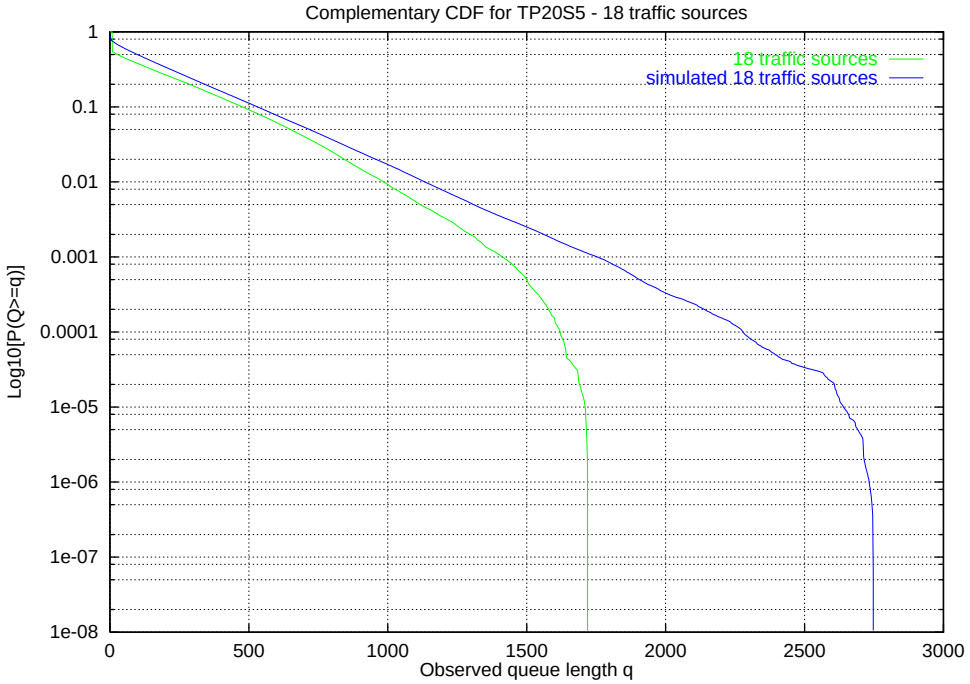


Figure 5.4: General experiment trend of Complementary CDF of experiments using the TP20S5 traffic source.



(a) 15 sources



(b) 18 sources

Figure 5.5: Complementary CDF of experiments using the TP20S5 traffic source.

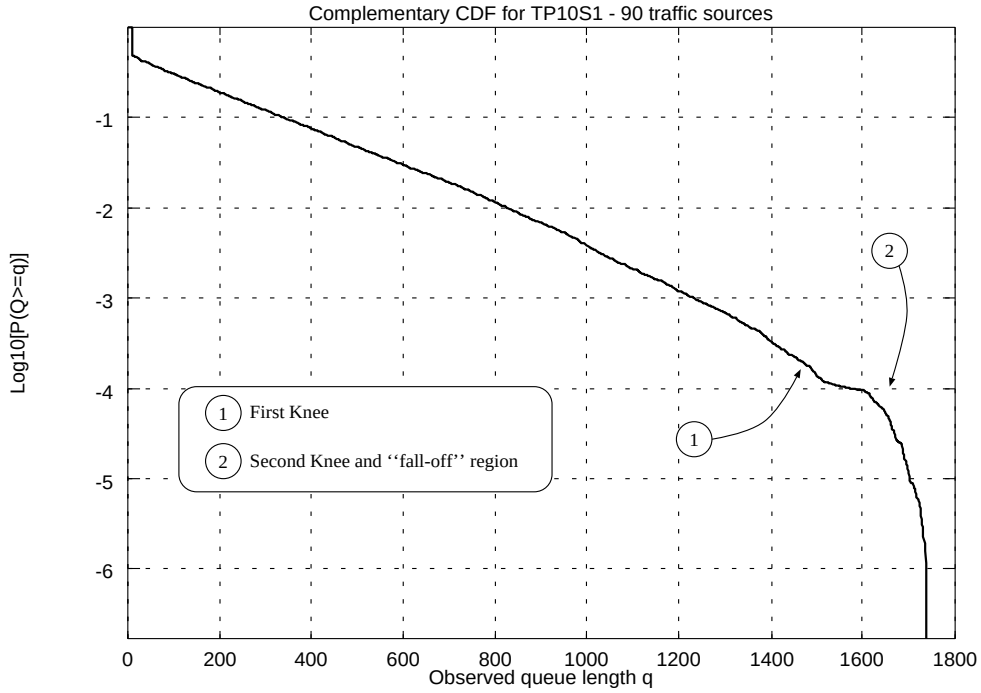


Figure 5.6: Complementary CDF of one experiment using 90 traffic sources of the TP10S1 traffic type.

5.2 Follow-up results

Figure 5.6 graphs the Complementary CDF resulting from a single experiment made using 90 traffic sources of the TP10S1 traffic type. The Complementary CDF has a double bend that is a characteristic to a greater or lesser extent of the Complementary CDFs of all the experiments conducted. Figure 5.6 labels the two knee bends, the second "knee bend" is a strong characteristic of each Complementary CDF and incorporates a drop-off in the $P(Q \geq q)$ values shown in the function. The first "knee bend" may only be an artifact, perhaps as a side effect of the fall-off, however it too is a common characteristic of the Complementary CDF.

The relationship between the position of the fall-off and parameters in the experiment were considered of interest to the evaluation and resulted in a round of explicit follow-up experiments. Potential relationships were tested by varying:

- the mean burst length,
- the total number of cells transmitted, and,
- the size of the working set used by each physical Fairisle port generating cells.

The working set used in an experiment is the size of the ICT list used in a physical Fairisle port to control the rate at which cells are transmitted. The number of ICT values

that can be contained in a single Fairisle port is limited by the amount of memory the port contains; for these experiments the limit was 1.4×10^6 ICT entries. In this case the working set size is 1.4×10^6 ICT entries. Additionally experiments were also run with the seed of the random number generator used in the theoretical sources varied to see the impact on the resulting Complementary CDF. Table 5.2 lists the parameters of each of the follow-up experiments along with the figure number in which the resulting Complementary CDF is graphed.

Variable	Default Value	Figure showing Complementary CDF
MBS	25 cells	5.8(a)
Run length	1×10^8 cells	5.8(b)
Working set size	1.4×10^8 cells	5.9(a)
Random seed	–	5.9(b)

Table 5.2: Initial experiment parameters and figure numbers graphing the resulting Complementary CDF relation.

Figure 5.8(a) graphs the variation in the Complementary CDF as a result of changing the mean burst size of the theoretical sources. The TP10S1, the theoretical source used throughout this study has a default MBS of 25 cells; Figure 5.8(a) shows the result of using an MBS of 2 cells and an MBS of 50 cells in addition to an MBS of 25 cells. The Complementary CDF graphed in Figure 5.8(a) does have a consistent variation in the drop-off point, however the rate of decay in the exponentially decaying relationship between $P(Q \geq q)$ and the observed queue length also varies dramatically between values of the MBS. Figure 5.8(a) enables us to confirm that MBS variation would cause dramatic variation in the Complementary CDF, however not in a manner that maintains a constant rate of the exponential decay relationship between $P(Q \geq q)$ and the observed queue length.

Figure 5.8(b) graphs the Complementary CDF resulting from experiments were the total number of cells in the experiment were varied. The total number of cells in each experiment was selected to be 1×10^8, this graph shows results using this value as well as several values larger and smaller. Ideally the largest value possible is the best to use, however the total number of cells controls the lifetime of the experiment. An experiment made using 1×10^8 cells runs for 2 hours, an experiment made using 1×10^{10} runs for approximately 8 days and 8 hours. 1×10^8 cells was selected as the best balance between the lifetime of the experiment and the statistical accuracy of the results. It should be noted that the drop-off point appears to be 'bounded'; the 1×10^{10} cell run presents a smooth Complementary CDF that has a drop-off right in among the drop-off of the Complementary CDFs of the other experiment lengths. There are several other points worth noting in the relationship between the drop-off in the Complementary CDF and the number of cells in the experiment. Firstly experiments of too small a length will produce a Complementary CDF that has a drop-off to a much earlier value of the observed queue length (for example

the Complementary CDF for the experiment made using 1×10^6 cells); this early drop-off may be due entirely to there being too small a sample of cells used in the building of the Complementary CDF at large values of the observed queue length. The other note is the considerable variation in the drop-off between experiments that are made using the same number of total cells, (compare the two runs of 1×10^9 cells). One conclusion is the experiment length is certainly a factor, however beyond a given point it has little effect on the drop-off with other factors playing a more significant role.

The working set size was the other experiment parameter that may have had a significant role in the point at which the drop-off occurs as the Complementary CDF tends towards an insignificant value of $P(Q \geq q)$. Figure 5.9(a) graphs the Complementary CDF resulting from experiments where the working set size is varied. From this graph it is clear that the working set size plays a significant role in the point at which the fall-off will occur in the Complementary CDF. A set size of 35,000 cells has a fall-off at an observed queue length of around 320 whereas experiments with a working set size of 1.4 million cells have a fall off at observed queue lengths values as high as 1,020. This set of results made it clear that maximising the working set size was also going to be an important issue in any CAC test environment. The results of Figure 5.7 become clearer in the context of increasing the working set size although there was still an anomaly in Figure 5.7 where decreasing the working set used in each physical Fairisle port by using 8 physical ports made no significant difference in the fall-off point of the resulting Complementary CDF.

The reason the working set size has such a significant effect required further explanation. The number of ICT values that can be contained in a single Fairisle port is limited by the amount of memory the port contains; for these experiments the limit was 1.4×10^6 ICT entries. As a result, an experiment that requires the transmission of 1×10^8 cells will need to transmit the contents of a single Fairisle port, a single working set, 72 times. To understand why this could be a problem, it is worth reiterating the method by which the ICT entries are generated. Section 4.2 covers this topic in greater detail but in summary: a single Fairisle port transmits the cells of multiple theoretical sources, a single Fairisle port transmits an ICT list that has been created off-line by multiplexing together, off-line, the output of multiple theoretical sources. This will mean that for one pass through the ICT list the theoretical sources will be independent and have no correlations however each repeated transmission of the ICT list will increase the correlation of the total traffic stream. This correlation results because the multiplex of the independent traffic sources is 1.4×10^6 cells long however over 1×10^8 cells of transmission the pattern of the multiplexed traffic is repeated 72 times.

This issue of correlation in the ICT data stream presents two solutions. The first solution is to use Fairisle ports with a larger capacity, ideally the port could be set to contain the whole 1×10^8 ICT entries. This solution would be ideal however the Fairisle ports, being designed as parts of an ATM switch system and not as traffic generators, do not have the ability to be equipped with the memory[5] required for such a solution. The second solution is to use more than one Fairisle port to transmit the stream of cells and to

[5] 2×10^8 (190 Mbytes) of memory for the trace alone.

multiplex this stream of cells prior to it entering the buffer of the ATM switch. To make full use of this solution ideally we would use 72 Fairisle ports each transmitting the cells of 1/72nd of the total cell count, the cells from each switch would then be multiplexed together entering the ATM switch for measurement. This solution is not available as we do not have 72 Fairisle ports; however a partial solution becomes available.

The number of times the working set in each Fairisle port will be repeated can be reduced by using more than one Fairisle port to transmit the total ICT trace. While using one Fairisle port to transmit 1×10^8 cells will require the 1.4×10^6 ICT entries of a single port to be repeated 72 times; the contents of a single port will only need to be transmitted 36 times if we use two Fairisle ports, 18 times for four Fairisle ports and so on. It needs to be remembered that the ICT trace used in a single Fairisle port is the multiplex of the output of an arbitrary number of theoretical sources. This presents a simple technique for reducing the number of times the working set of a Fairisle port is repeated.

In order to reduce the number of times the working set of a single Fairisle port will be repeated we could increase the size of the trace created from the multiplex of all the theoretical sources. If two Fairisle ports were used, the trace size could be 2.8×10^6 ICT entries in length, each port would hold half the trace and the whole trace could be transmitted by using each Fairisle port in order. This is quite a complicated solution requiring coordination between the two or more Fairisle ports. An alternative is to remember that the ICT trace transmitted by the Fairisle port has been created by multiplexing an arbitrary number of theoretical sources together.

Instead of using each Fairisle port in sequence to transmit a longer trace, we could represent half the number of multiplexed theoretical sources in each ICT list. This would mean that twice as many cells (one cell per ICT) were used to represent the same number of theoretical sources; thereby increasing the working set size used to represent the total number of cells in the system. In effect we have doubled the number of Fairisle ports being used to transmit the total trace; as a result the relation mentioned above still holds to transmit a total of 1×10^8 cells: 1 Fairisle port will need to transmit its ICT list 72 times, 2 ports will need to transmit their ICT list 36 times, 4 ports – 18 times and so on. This whole technique works by each physical Fairisle port transmitting the multiplex of fewer theoretical sources.

Figure 5.7 graphs the effect this technique has on the resulting Complementary CDF. A fixed number of theoretical sources (92) are distributed among a variable number of physical Fairisle ports. Tests were made with 2, 4 and 8 physical Fairisle ports. Figure 5.7 results show that a clear upper boundary on fall-off point is present at an observed queue length of around 1840 cells. This set of results was used to justify the decision to use 4 physical Fairisle ports for the majority of experiments; increasing the number of generators did not appear to make any difference in the fall-off point of the graph.

The final follow-up round of experiments conducted as part of the initial experiment phase were to conduct several identical experiments with only the seed of the cell generator varied between runs. The seed controls the random number generators used in the theoretical 2-state Markov on-off cell sources. Only simulated results are shown, in this way all variations of the experimental rig such as working set size have been eliminated.

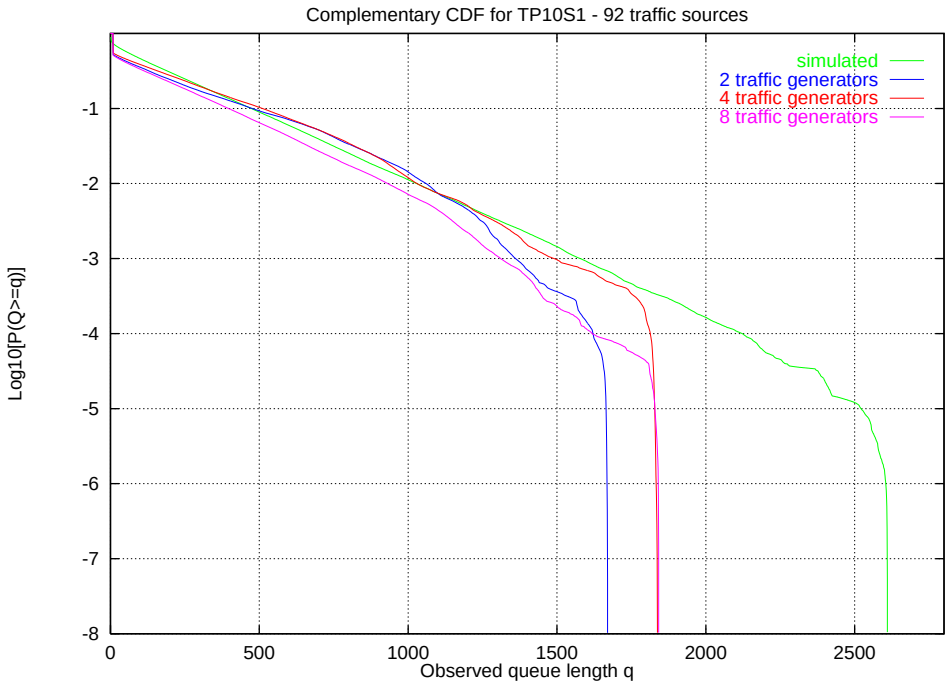
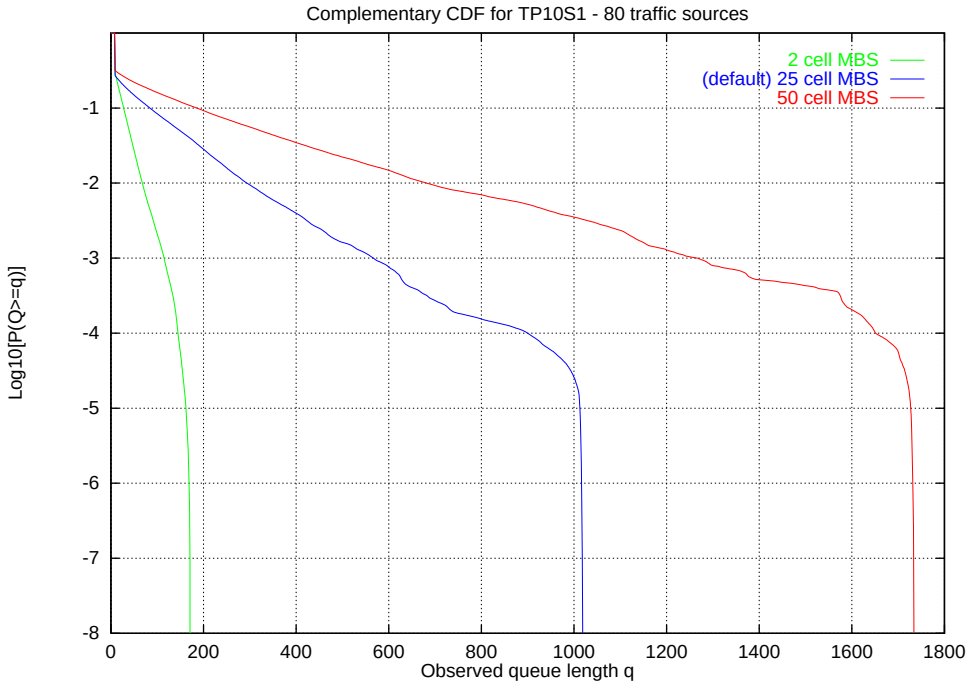
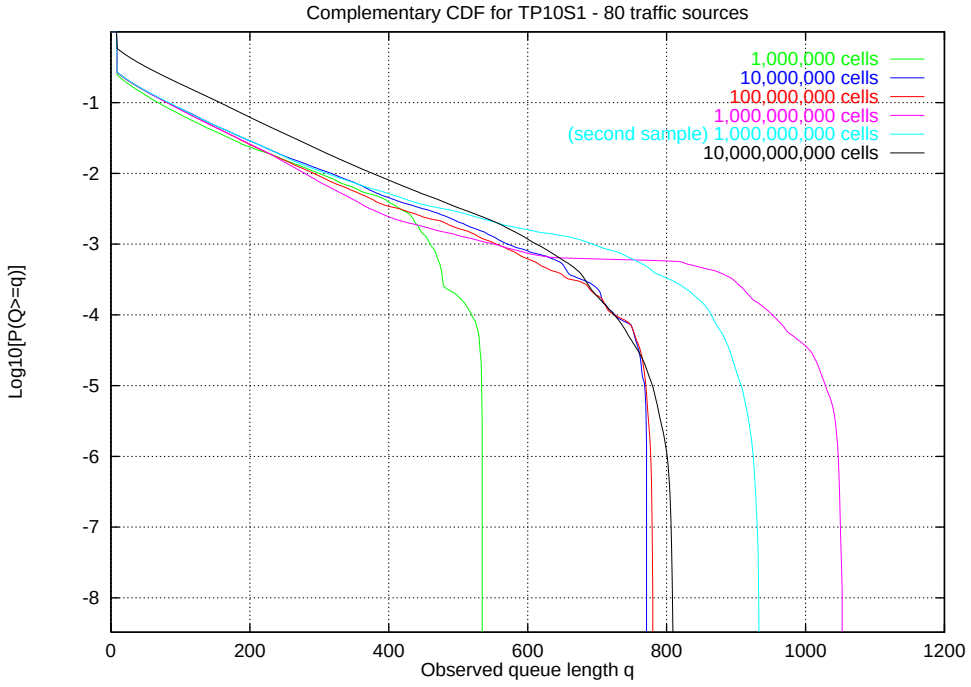


Figure 5.7: 92 sources with a variable number of physical generators

The resulting Complementary CDFs created using several different seed values are shown in Figure 5.9(b). From these results the value of the observed queue length at which the Complementary CDF drop-off occurs varies between 700 and 965. The conclusion from this is that significant variation in the drop-off occurs purely as a result of variation in the multiplexing of data into the ATM switch. Such a result has meant that while other causes of variation in the shape of the Complementary CDF can be noted, considerable variation in the Complementary CDF could be resulting from the random multiplexing of sources at the switch, an uncontrollable variable in the experiment.

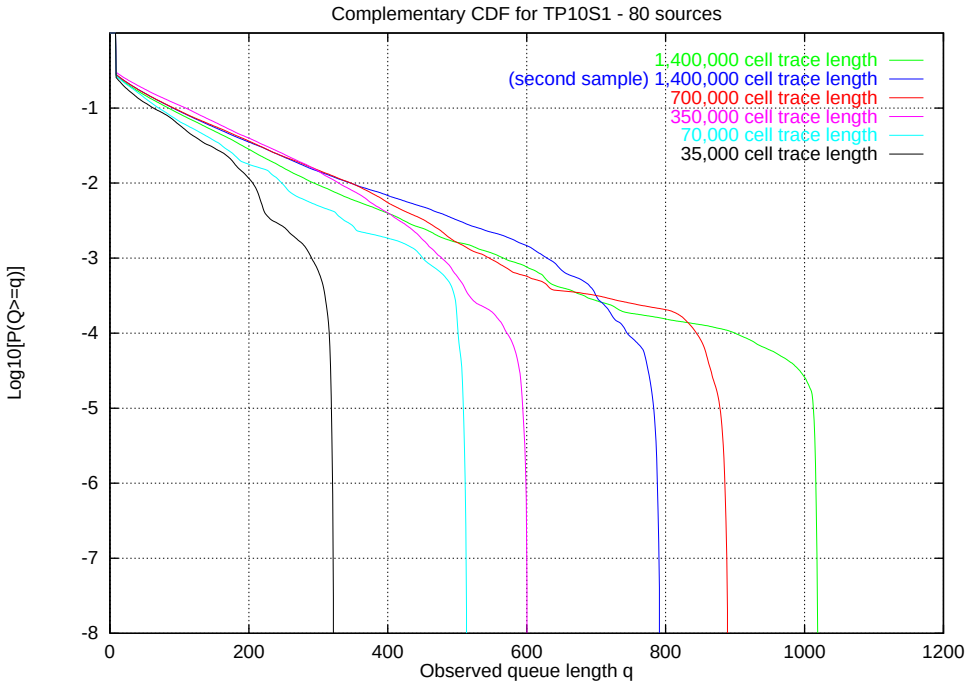


(a) Variable burst length; the default MBS is 25 cells.

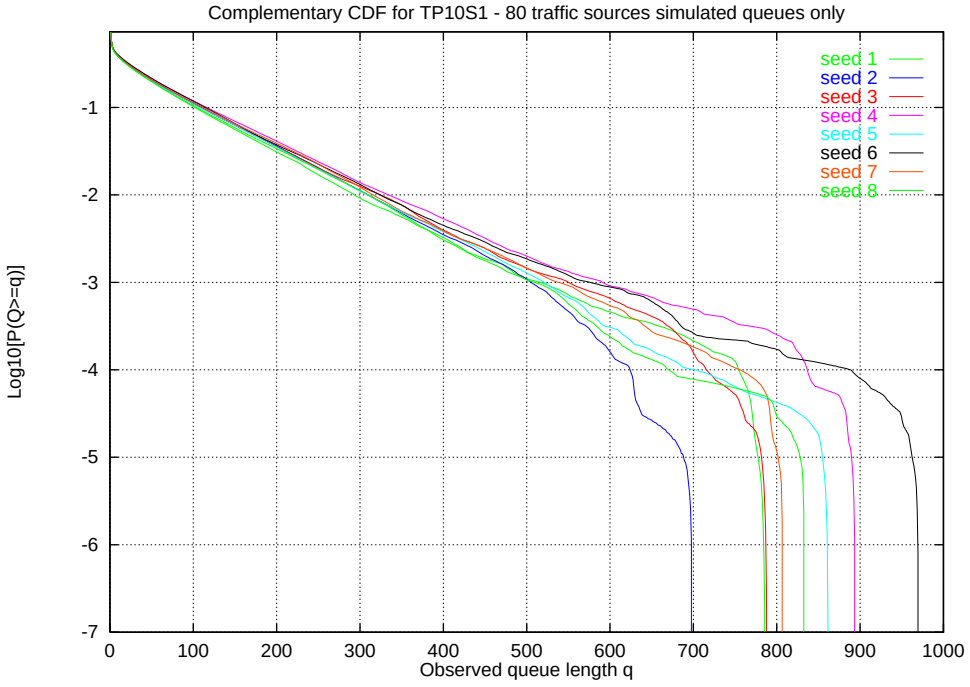


(b) Variable length of experiment run; the default run length is 1×10^8 cells.

Figure 5.8: Complementary CDF of experiments with 80 TP10S1 traffic sources.



(a) Variable working set size; the default working set size is 1.4 million cells



(b) Variable seed used by the cell generator; simulated results only.

Figure 5.9: Complementary CDF of experiments with 80 TP10S1 traffic sources.

6 CAC Evaluation Environment

The experimental configuration used in the evaluation of the BT adaptive Threshold based CAC algorithm is described in this section. The philosophy used in the design of the CAC test-rig was to allow comparison of one CAC algorithm with another under near identical conditions of connection load. This implies that the evaluation environment must allow control of the traffic types carried by each connection as well allowing the number of connection attempts, connection holding times and the respective distributions of MCAR and MCHT to be configured. In addition to the connection load, the test-rig must allow substitution of one CAC algorithm for another while allowing information about the performance of the system to be collected and compared. This led to three primary desires in the evaluation environment; firstly the test-rig had to allow good performance, allowing a high rate of new connections to be attempted per second allowing comparison with commercial implementations and theoretical simulations, secondly the test-rig had to be adaptable – allowing variations in the connection load, CAC method, and measurements made of the system both for each CAC to make use of as well as allowing the performance of each CAC to be tested. Finally the evaluation environment must have high repeatability of experiments, this is considered essential if the results obtained using different CAC algorithms are to be usefully compared.

In order to build the test-rig one issue in particular from the initial experimental results needed to be addressed: this was the problem of traffic generators. Traffic generators under the regime where traffic generators are based on Fairisle ports meant limits in the length of the trace that could be played. This limit in the working set size was shown in Section 5.2 to be a potentially serious problem in the behaviour of the system: multiple connections exhibited self-similarity as the connections were held in lock step over multiple transmissions of the ICT list. As part of the process of creating a working CAC test-rig, the creation of new method of traffic generation was needed and the solution found discussed here in this section.

An evaluation of the new test-rig is given in this section; this evaluation looking at the CAC test-rigs' performance, repeatability and adaptability. Performance aspects required the greatest single attention in order to achieve acceptable levels of connections able to enter the system per second. Once a satisfactory performance level was achieved, repeatability was assessed. Adaptability of the system is shown in the variety of mechanisms that have already been incorporated into the system; mechanisms that include CAC algorithms, measurement techniques and numerous aspects of new connection types and arrival mechanisms.

6.1 Test-rig construction

The CAC test-rig consists of a combination of hardware in the form of the ATM switch and ATM interface cards, and software in the form of software to generate new connections, perform CAC operations, obtain measurements from the ATM switch, generate traffic sources and control the generation of traffic sources. Figure 6.1 shows the implementation

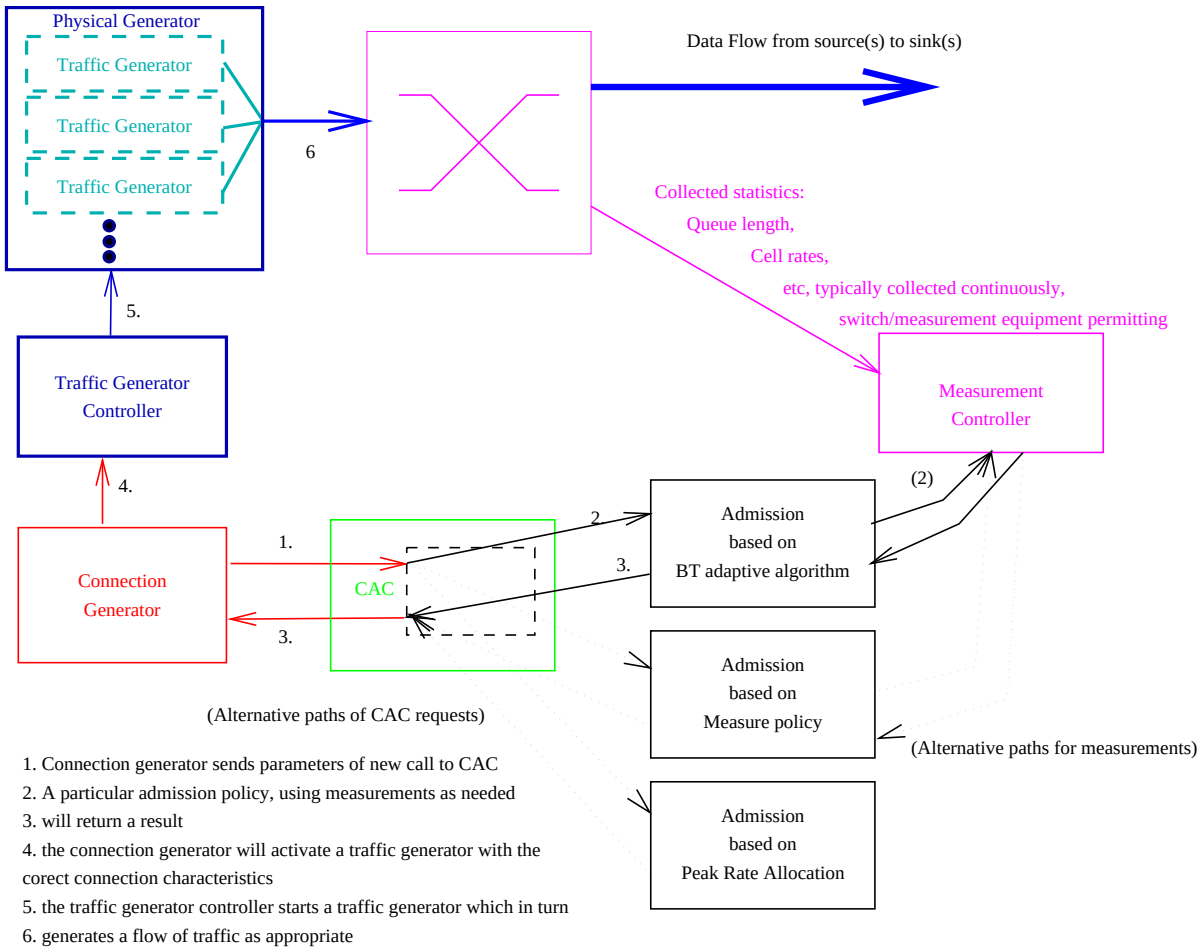


Figure 6.1: Architecture for the implementation of a test environment to evaluate CAC mechanisms.

architecture adopted to evaluate CAC algorithms in general and the Key CAC algorithm in particular. In this section specific components of the CAC test-rig are discussed followed by the procedure used when a new connection enters the system.

6.1.1 ATM switch - Fairisle

The switch used in the CAC test-rig at this time is the Fairisle ATM switch discussed in Section 4.1. The Fairisle ATM switch enabled retrieval of measurements of a quality suitable for use in the CAC algorithms. Additionally the Fairisle ATM switch is able to make measurements of the switch buffers, cells counts and cell timings enabling an assessment of the performance of the CAC algorithm. The Fairisle ATM switch operates in an environment that is interfaced to with a measurement controller.

6.1.2 Measurement controller

The measurement controller, a process running under Unix, obtains from the switch the measurements that may be required as input into a CAC algorithm. Additionally, the measurement controller measures not only the traffic activity, is an input to the measure algorithms, but also the QoS experienced by the traffic – its CLR, queue length distributions, inter-cell loss times and other measures which permit off-line comparisons between the performance of the system in operation and the observed performance of a simulation using a simulated switch model.

While the measurement controller performs important functions in the retrieval and storage of measurements from the switch, the server's most important tasks are the matching of asynchronous measurement requests (from the CAC algorithm) to the synchronous methods in which measurements must be taken. The measurement controller also interfaces between the proprietary inter-machine protocol used by the switch with a standard RPC interface that is used between the measurement server and the CAC system itself.

6.1.3 Traffic Generator

An ideal traffic generator for the CAC test-rig would be able to represent the traffic of an arbitrary number of connections. Each connection would be carrying traffic of an arbitrary number of different traffic types. The traffic generator should be controllable so that the traffic source representing any connection can be arbitrarily turned on or off, thereby starting or stopping the cells of that connection.

The evolution started with the traffic generator of Section 4.1.1, based on a Fairisle ATM port, first with the need to increase the size of the ICT memory space. The number of ICT entries on the Fairisle ATM port is limited to 1.4×10^6 cells. By designing the new physical generator on a standard PC the working set, the number of cells in a single ICT list, can be increased dramatically. While an ICT list that contains a list of cells for the full experiment (1×10^8 cells) is not possible (it would require 2×10^8 bytes of memory, or 191Mbytes) the working set can be increased by at least an order of magnitude over the capacity of the traffic generator based on the Fairisle ATM switch. This however does not form an ideal solution.

An improvement on this revision of the physical generator is to have the ICT list represent the cells of only one connection. The physical generator has multiple traffic generators running, each reading from a random start point in the ICT list, for the duration of that particular connection. Figure 6.2 shows how four connections can be simultaneously in progress, the ICT values from the same ICT list are read and cells are emitted at the noted rate. This can be compared with Figure 4.2 of Section 4.1.1; the original physical generator had a multiplex of traffic pre-loaded, the new style physical generator can multiplex ICT lists dynamically. The cells from each of the (four) connections are multiplexed together and the multiplex of cells are emitted from the generator. The start and stop position of each connection is independent, each position based on a random number that is part of the new connection process, not the cell generation process. Because the multiplex of con-

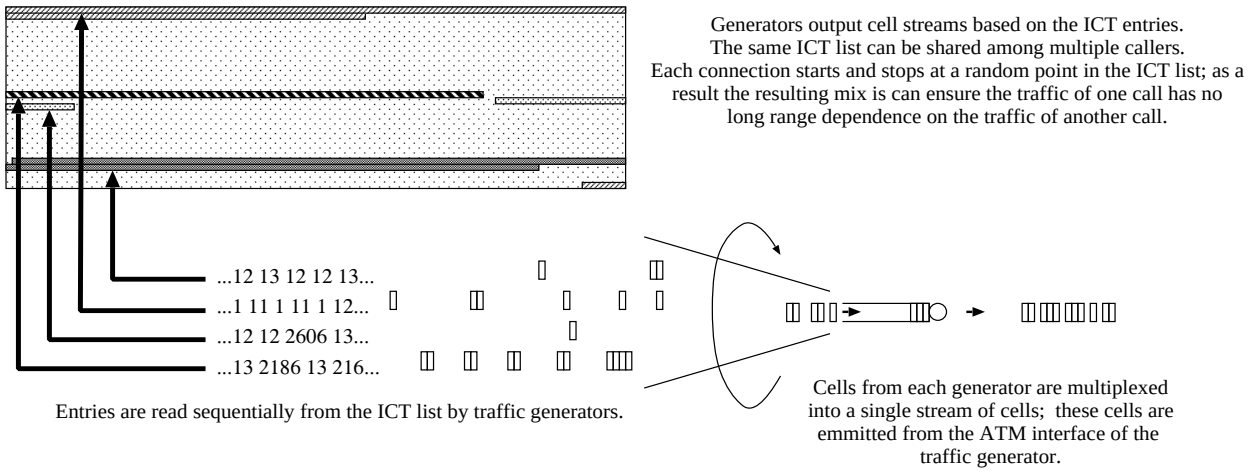


Figure 6.2: Generating the cells of multiple connections by multiplexing the ICT list of a single connection.

nections is random, dependent only on the admission of new connections the cell generation system gives long term independence between the connections added to the advantages of a larger ICT play list. In addition to playing pre-generated theoretical sources, this system makes it possible to play pre-generated, or recorded traces of cell traffic from other traffic types such as video stream traffic. The PC-based physical generator can offer additional advantages in the production of cells from theoretical traffic types.

An added improvement offered by the PC-based system of Figure 6.2 is the the real-time generation of the ICT stream of each connection. The additional CPU power available on a PC-based physical generator (as compared with the CPU power available on the Fairisle ATM port based traffic generator) has meant that the production of traffic streams for theoretical traffic types can be performed in real-time. In this way all the tasks of the original off-line ICT list generation (Figure 4.3 in Section 4.2) have become part of the new physical generator. Figure 6.3 shows how the tasks of generating independent traffic streams for each connection and then multiplexing the resulting cells into one output stream are performed by the new physical generator. It should be remembered that this physical generator overcomes any limitations in working set size that occurred in the original physical generator based on the Fairisle ATM port. Investigated more fully in Section 5.2, limitations in the working set size of ICT entries created off-line would suffer long term dependencies and cause distortion of results in the switch. A theoretical generator generating ICT timings for a cell stream in real-time could run indefinitely, this means the working set size restriction in the physical generator is removed completely; memory no-longer being the limiting factor. The result at this stage is a dynamically controlled physical generator able to generate a multiplex of traffic based on theoretical models (such as Markov 2-state on-off) each connections' traffic source independent and each source able to generate cells for indefinite periods of time. Such a system is almost ideal for the needs of the CAC

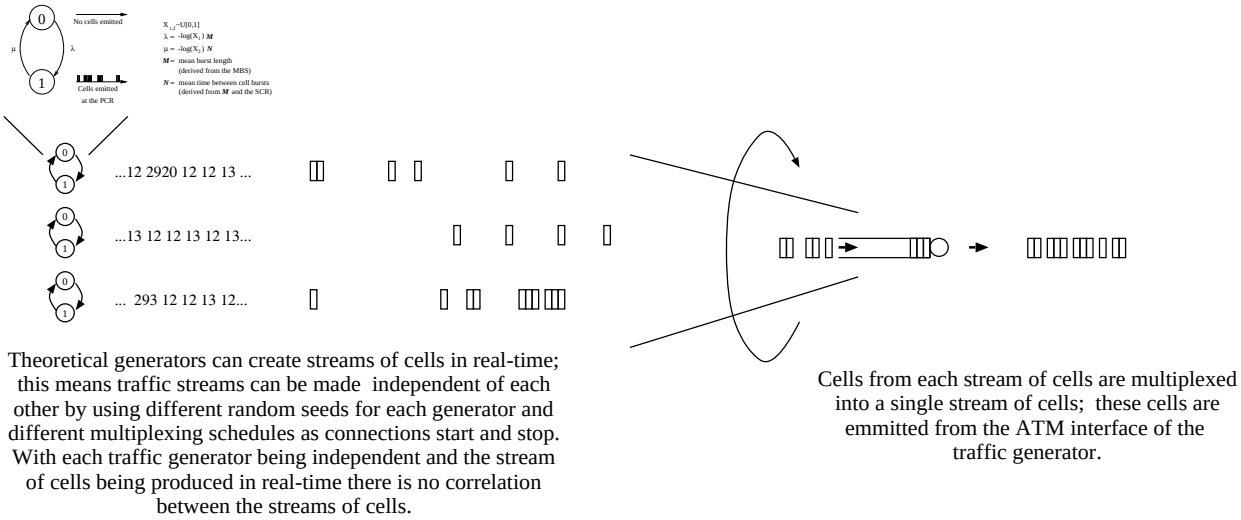


Figure 6.3: Generating the cells of multiple connections using multiple theoretical generators to create cells in real-time and multiplexing the resulting cells into a single stream.

test-rig.

The area the revised physical generator requires in addition to theoretical generators is the ability to send cells that are the result of a converted video stream such as VP10S1 or VP5S2 of Section 2. Either of the video streams VP10S1 and VP5S2 can be represented as a set of ICT entries. This situation is different from the list of ICT entries used to represent a theoretical trace because there is an upper limit on the size of the trace enforced, there is an upper limit on the size of the original video stream. This means a physical generator of the style of Figure 6.2 can be used. This generator which multiplexes together the cells from an ICT list that represents the traffic of only one connection. Such ICT list-based generators can then create cells for arbitrary time periods representing each connection while its in progress. A hybrid physical generator can perform all the traffic generation requirements in one single box.

Figure 6.4 shows how a merger of Figure 6.2 and Figure 6.3 can be used to create a physical generator able to transmit a multiplexed stream of cells from theoretical generators creating cells in real-time and from generators reading from a list of ICT entries. In this way the working set size limitations that has previously affected theoretical sources is removed and the flexibility of being able to generate off-line lists of ICT entries, to represent video streams, is retained. In addition to being able to deliver cell streams consisting of all required traffic types, this revised physical generator can be dynamically controlled, able to stop and start individual traffic sources using a purpose built RPC mechanism.

The whole physical generator runs on a PC that is running the Nemesis Operating system [25] which allows the construction of complex, time-critical tasks (the real-time creation of traffic traces) and the timely operations of device-drivers. A special purpose

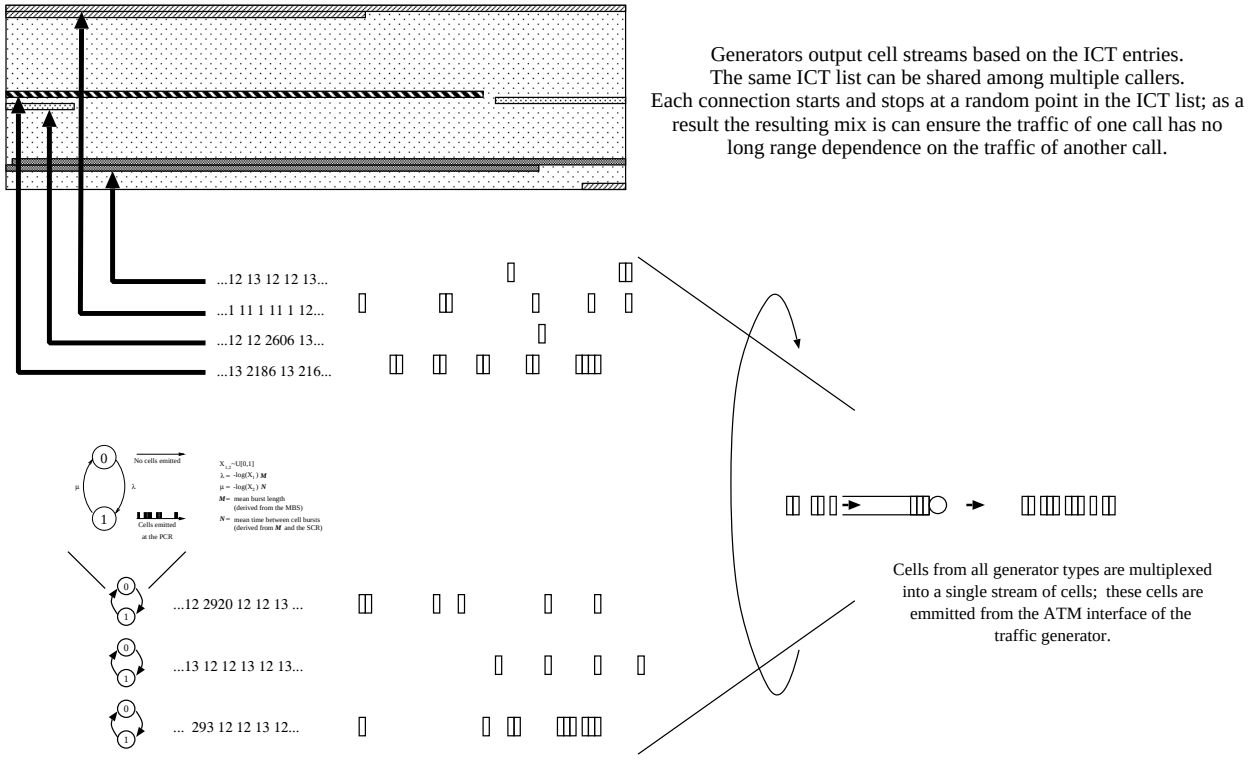


Figure 6.4: Hybrid physical generator able to create cells from theoretical traffic sources operating in real-time and from an ICT list loaded into memory.

device driver was written for the physical generator. Nemesis can make timely guarantees to the device driver so that batches of cells to be transmitted are not delayed. In addition, guarantees of timeliness can be made to the RPC based control mechanism to ensure time-bounded actions and replies.

With an individual traffic generator representing each connection in progress, the physical generator is capable of saturating the ATM transmission link should this be required and the ability to combine a virtually unlimited number of traffic types of both the theoretical generator or those based upon lists of ICT entries gives us unrivalled flexibility in experimentation.

6.1.4 Traffic generator controller

The traffic generator controller, a process running under Unix, will instruct the traffic generator to start and stop individual traffic sources representing each connection as these connections are set-up and pulled-down. The traffic generator controller, like the measurement controller, gives an interface between the standard RPC based interface that is used between itself and the CAC system; interfacing to the purpose-built inter-machine protocol used by the traffic controller. In later revisions of the physical traffic generator it is expected to be able to dispense with the traffic generator controller altogether. This would be made possible by the physical traffic generator having a standard RPC based interface.

6.1.5 Connection Generation

The connection generator, a process running under Unix, will initiate new connection attempts into the CAC test-rig. Connections entering the system can be described by the parameters for the arrival rate of new connections, the connection holding time and the traffic each new connection will carry. The parameters of connection inter-arrival rate and connection holding time can have specified values or have a range of values based on a distribution – for example the period over which a connection will be in progress could have an exponential distribution with a given mean. The value for traffic type would typically be specified for a set of connections. The three parameters can also be specified in a file; in this way logs of connection events can be processed to produce a set of connection arrival-rates, durations and traffic types.

The connection generator is able to generate new connections of more than one traffic type simultaneously. For example a connection carrying a particular video stream could be being attempted every 10 seconds, while connections carrying a theoretical traffic source are being attempted with an MCAR of 55 seconds. As a result of this flexibility a wide mix of new connections and of traffic types entering the test-rig can be achieved.

6.1.6 CAC and admission policy

The CAC component forms the core of the CAC test-rig. The CAC component has the capacity to change the CAC admission policy as required. Only one policy is in place during

any experiment however consecutive experiments can operate with only the admission policy itself or the control parameters of any particular policy being changed.

During the generation of new connections the traffic type and the parameters that describe traffic that the connection will carry are declared to the CAC algorithm. The parameters of each new connection can be specified in any of the TM 4.0 parameter formats [1]. In this way each new connection can specify its PCR, SCR, IBT and any other parameters for full ATM Forum compliance. Each new connection presents its parameters to the CAC system and requests a connection to be set up across the switch.

Each admission policy obtains the required measurements from the measurement controller as part of that particular CACs' decision process. Each admission policy can obtain the measurements of the type and format it requires, in the case of the BT adaptive CAC algorithm measurements are of instantaneous line utilisation while for Peak Rate allocation no measurements are required as an input into the CAC algorithm.

6.1.7 Cell time-frame scaling

Section 4.1 discusses, the Fairisle ATM switch used in the CAC test-rig. In that section it is noted that the rates of traffic sources are scaled by a factor **D**, this factor is a multiplier on the time between cells. As a result, the passage of time on the network, and hence the passage of time in the experiment as a whole, has been slowed down by the factor **D**. There is a single drawback to this system – experiments run **D** 'times' longer because all values of time in the system are scaled up by **D**. For example a connection with a holding time of 10 seconds will from the experimenters perspective have a duration of **D**×10 seconds. Such a system of scaling has the immediate effect of allowing all parts of the test-rig to do more processing for the each cell, thereby giving **D** times the amount of time to do processing required per cell (and per connection and per experiment). Such extra time is important when the switch fabric is required to perform numerous timing and counting operations on receipt of each cell. This technique of time scaling has been used in several Fairisle projects, most significantly Crosby in [11], and as previously reported to BT in [10].

Throughout this document all times stated for kit performance, connection setup, connection holdings periods, measurement period, and any other time frame in the experiment are given in unscaled time; that is time that has not been multiplied by **D**. Using measurements on this time scale make reported experimental results directly comparable with measurements made on other systems.

6.2 Test-rig operation

The CAC system works as follows: a connection generator (running on Unix) is responsible for 'generating' according to some distribution. The connection arrival times could also be from a previously collected trace of measured arrivals, the arrival of new connections. New connections may be of multiple types, and each connection may, according to a random distribution, determine its connection type and any set of parameters which it is required to present. In the case of the adaptive CAC algorithm evaluated, none of

the new-connections' traffic parameters are used, however for full ATM Forum compliant connections, a connection might be required to declare its SCR, IBT and any other useful parameters. New connections generated by the connection generator arrive at the CAC decision system when they are generated. Each connection presents its parameters to the CAC system and requests a connection to be set up across the switch. The currently loaded CAC algorithm, using measurements from the switch of the current activity, makes a decision as to whether or not to admit the connection. Only one CAC algorithm operates in any one experiment.

If a connection is admitted, the CAC algorithm will reply to the connection generator accepting the connection. The connection generator then instructs the traffic generator controller to 'set-up' a new traffic generator with the appropriate parameters for a connection of this type. The traffic carried by this connection might be on-off, some other analytical model, or trace driven. The traffic generator controller then starts the new connection by instructing the physical generator to create a traffic generator with the correct parameters and to start the new generator. The cells of this new connection will then enter the multiplex of streams of cell that the physical generator is transmitting into the switch.

When each new connection is created, apart from its traffic type and arrival time, a new connection will have associated with it a lifetime, or connection holding time. This connection lifetime, like the arrival time, can be drawn from a theoretical distribution or a trace driven set of values (perhaps measured from a previous experiment). Once the connection holding time is reached the connections' traffic source is stopped and that connection is 'cleared down'.

It is important to stress that in this set-up there is no real ATM signalling. The processes running off-switch assume the full load of the 'signalling' and therefore it is possible to emulate the arrival of connections at rates far higher than could be sustained by any real ATM signalling implementation. Also, the setting up of a 'connection' through the switch is optimised in the sense that all VCI/VPI pairs to be used in an experiment are mapped through the switch before the experiment starts, meaning that there is no need to actually set up paths during the experiment. This greatly aids efficiency, and means that can be achieved realistic connection setup rates for large networks.

6.3 Test-rig evaluation

During the construction of the CAC test-rig, effort was made to ensure the goals of performance, repeatability and adaptability were being meet. The test-rig was required to achieve high rates of new-connection setup: the rate at which new connections could be attempted and started when the connection was admitted into the system. Section 6.3.1 covers both the connection performance of the test-rig and work performed to improve this performance. The performance of the test-rig in the minimum connection set-up period has meant it is possible to test CAC algorithms under a wide range of input conditions (such as MCAR and MCHT). The ability to conduct multiple experiments and obtain repeat results from each consecutive run was the next part of the test-rig evaluation.

The test-rig must allow high repeatability between experiments; ideally, consecutive

experiments with no parameter changes should yield as identical results. Section 6.3.2 reports results conducted to assess the repeatability of experiments on the CAC test-rig. As part of this assessment of the repeatability of experiments the results are summarised to give error boundaries and confidence intervals on results gained using the CAC test-rig.

The final phase of test-rig evaluation, Section 6.3.3, takes the form of a set of tables documenting the various new-connection regimes for controlling the connection admission rate and connection holding time, methods of measurement that are implemented or plan to be implemented along with the various CAC algorithms either implemented or being planned. The numerous connection regimes, measurement types, CAC policy implementations and traffic types show the CAC test-rig well satisfies its requirement as an adaptable platform for supporting CAC algorithm evaluation.

6.3.1 Performance

When a connection is entered into the system an assumption is that there is a negligible amount of time between the time when the new connection has been generated and, assuming acceptance, the moment when cells transmitted by the corresponding traffic generator will start entering the data stream. This assumption is not valid in anything other than a theoretical test structure. However, it is important to quantify and where possible overcome such a delay between a new connection entering the system and cells being produced by the system so as to minimise the impact of experimental effects being introduced into the evaluation experiments. In this way theoretical results and experimental results can be compared more closely.

The performance goal in the construction of the CAC test-rig was to reduce the new connection generation, new connection test and new connection start-up delays to a minimum; the smaller the delay achieved the closer to the values used in a naïve simulation. When compared to the real-world implementation, aiming for a minimal delay could be seen as unnecessary – several authors [2, 28] note that in commercial ATM switch systems the call setup process for a new connection can take 20–200ms. Such a quoted value for the delay does not include the additional time required for the end-system to become active. In the work to reduce the delay on the CAC test-rig, the delay includes all parts from generation of the new connection request through to the moment cells are emitted from the ATM interface of the traffic generator.

The delays in the pathway between the generation of a new connection request and the emission of cells into the ATM switch take several forms: firstly there is time taken in the execution of code on the various machines that the CAC test-rig runs, secondly there are delays related to the communications between components of the CAC test-rig and finally there are delays in the physical generator that will cause a delay between the starting of traffic generators and the emission of cells from the ATM interface and into the ATM switch.

The performance evaluation of the time delays through the CAC test-rig was hampered by the lack of a high resolution timer with consistent behaviour from one brand of Unix system to another. Quite commonly it was found that timers available on the system

would be being updated at a much lower frequency than stated in the documentation. This is a common problem when the update of clock counters is a schedule-able task in the main kernel of an operating system. While solutions such as [32] have been proposed and used successfully, there was neither the time nor the need to construct such a solution. A solution to the problem of unreliable timers used in this evaluation was careful monitoring of the timing machines and pre-testing of the machines to establish clock update behaviour.

The first cause of delays, the execution of code on the various systems that make up the CAC test-rig, is difficult to quantify or reduce substantially. During the flow of the CAC test-rig, from generation of a new connection attempt to the starting of a traffic generator 10,000 lines of 'C' code on Unix systems and 3,000 lines of 'C' and intel-assembler code on Nemesis systems is executed for each connection accepted into the system. There was no work directly spent on speeding up performance of the program code itself; it was thought the majority of the delay time could be attributed to inter-machine communications and that anything that reduced the delays in the inter-machine communications would more greatly reward time invested.

The second delay, is a compound of the many instances of inter-machine communications. In this version of CAC test-rig the inter-machine communications are in the form of Remote Procedure Call (RPC) operations. If as is the case for the BT adaptive CAC algorithm, the CAC algorithm requires an instantaneous measurement of traffic for each new connection on which the CAC mechanism must make a decision, the CAC code causes an RPC to be generated requesting the instantaneous value from the measurement server. The measurement server will in-turn request measurements from the switch. The results returned from the switch are then themselves forwarded to the CAC algorithm. If the CAC algorithm then admits the connection an RPC is sent to the traffic generator controller. The traffic generation controller converts this RPC into a request to the physical generator(s) causing the creation of a traffic generator with the traffic characteristics of that connections' traffic type. Replies and acknowledgements follow the reverse path. This involves a maximum of 8 RPC steps for a successfully admitted new connection.

Each of the 4 RPC links is based on an unreliable transport mechanism that will timeout and re-transmit the original request if data is lost. However, the RPCs are transmitted over a dedicated local Ethernet based network as a result because the chance of retransmission is random and dependent on the utilisation of the control channel because of this it can be difficult to bound the time taken in RPC communications. This was addressed by instrumenting the RPC mechanism so that retransmission requests could be recorded and an informed assessment made. The results from the instrumentation showed that over the course of the whole experimental evaluation (including the evaluation of other CAC algorithms detailed at the end of Section 7) the total number of retransmissions was noted to be less than one retransmission in ten complete experiments. This implies that for every 320,000 RPC calls (assuming about 4,000 accepted connections per experiment) there was only one retransmission. At this point the effects introduced by this aspect of the test-rig were discounted as negligible.

In order to reduce unnecessary RPC operations, an amount of caching was introduced. If the measurement server is called to give results for two consecutive connection admission

requests, before a new measurement could have been made, the earlier value is returned. This has no affect on the accuracy of the results because the measured value would not have been updated any faster than a nominated period. The effect of this caching was to reduce the total number of RPC calls made from 8 to an average of about 7.2; the exact value depends on the period over which the measurement of instantaneous traffic load was to be taken but the end results is not insignificant.

The other major change was to decouple the two RPC sides of the traffic generator. This meant that requests to turn on traffic generators were be answered 'instantly' by the traffic generator controller allowing the connection generator to continue with its various tasks. A separate sub-process on the traffic generator controller would check routinely for changes in the status of generators (as requested by the RPC calls from the call generator) and would perform the required proprietary RPC call to control the traffic generator(s) on the Nemesis box. This had the advantage of reducing further the number of RPC calls made from 7.2 (average) to 5.2 (average) in the pathway traversed between the generation of a a new connection and a traffic generator being initiated. As a result, delays of subsequent operations, such as other new connections, would also be reduced.

This technique of decoupling the RPCs needed for traffic generator control has a drawback; a delay is introduced between the moment a new connection being accepted and the generator being started. The effects of this delay have been reduced by ensuring that the controller polls for changes (such as new connections) as fast as possible. In order to ensure that the connection is delayed only in time and not shortened, the turning off of the same connection is subject to the same delay (the original start-up delay is measured and taken into account).

A final solution for eliminating the extra delay of either poll times or RPC transmissions between traffic generator controller and the physical generator would be to remove the traffic generator controller entirely from the system. This has not yet been implemented as the various roles it performs have not entirely been eliminated – this elimination is the long term objective. The same solution is not available for the measurement controller. One of the reasons for this is that the measurement server cannot be run on an architecture similar to the connection generator and CAC components; the RPC interface to the Fairisle switch is alien when compared with available programming interfaces and as a result the measurement controller tasks are not easily assimilated.

Timely measurement taking is important both for measurements that are input to the CAC algorithms themselves and for auxiliary measurements that will be used to appraise the performance of the CAC algorithm under evaluation. The evaluation environment was designed to enable accurate measurements to be taken during the course of experiments – desirable measurements could include cell counts at the granularity of the measurement period for the instantaneous traffic load, queue contents (and length) information, queue discard information and utilisation measurements.

While it has been established that the Fairisle switches are capable of high resolution measurement without significant error, delays in timing can be introduced into measurements in the same way that delays could be introduced into control of the traffic generators. Such delays in auxiliary measurements are not significant in this set of experiments. How-

ever, such delays may introduce a significant error for CAC algorithms, such as the BT adaptive algorithm, that require timely ('instantaneous') information on utilisation of the link and so on. A partial solution was achieved, as mentioned above, by introducing a level of result caching to eliminate unnecessary calls to the measure event system. Apart from this technique, few other options were available.

The final delay, the value arising in the transmission system, arises from the *staged* generation mechanism whereby cells created by an instance of a traffic generator can take up to 800 cell times before they actually exit the interface card. The maximum error of $286.2\mu s$ only occurs as a one-off start-up delay for the connection. The figure is a maximum and the mean figure is much smaller. Additional alterations to the transmission stream occur in the form of jitter. Jitter characteristics are introduced as a result of commercial ATM switch infrastructure (Fore Systems' ASX-200 ATM switch) being used to perform protocol matching between the traffic generators (using 155Mbps Sonet interfaces) and the Fairisle ATM switch (using 100Mbps TAXI interfaces). The ASX switch is in link **6** of Figure 6.1. It is important to note the switch does **not** perform rate reduction tasks – cells are not generated at a rate greater than the line rate capacity of the Fairisle interfaces (100Mbps TAXI). Apart from protocol matching, the commercial (non-Fairisle) ATM switch infrastructure plays no other role in the ATM test-rig.

Following on from the description of the causes of delay, the delay values were quantified. Experiments established that delays between a new connection arrival and the start of transmission of its cells from the corresponding generator was on the order of 8.38 milliseconds. The statistics and distribution of this delay are shown in Table 6.1 and Figure 6.5 respectively.

Such a startup delay has an immediate effect on consecutive connection attempts. Connections cannot be attempted into the CAC and (where appropriate) started at a rate any faster than ~ 119.3 connections per second of the experiment. As the experiment has a mean connection arrival rate of 10 connections per second, this means that, with 95% confidence, about 0.27 % of connections may be affected. However, the figure is actually much less than this because the full delay impact is incurred only on new connections that are accepted into the system.

For new connections that are rejected from the system, the delay is substantially smaller. Additionally, the figure of ~ 119.3 new connections per second implies every connection was successful and required a traffic generator to be started up, this is not the case however the figure of ~ 119.3 new connections per second is a useful worst case boundary. The value of ~ 119.3 new connections per second being a maximum for the number of new connections that will be able to be made within this period implies a call set up time of 9ms. When compared with the earlier stated values for commercial call setup of 20–200ms [2, 28] – a call setup period of less than 9ms is quite acceptable. In real terms this means that, in the worst case, for an experiment of 6000 connections, less than 18 of those connections will be affected by system delays where the delaying of one connection will delay the next connection entering the system.

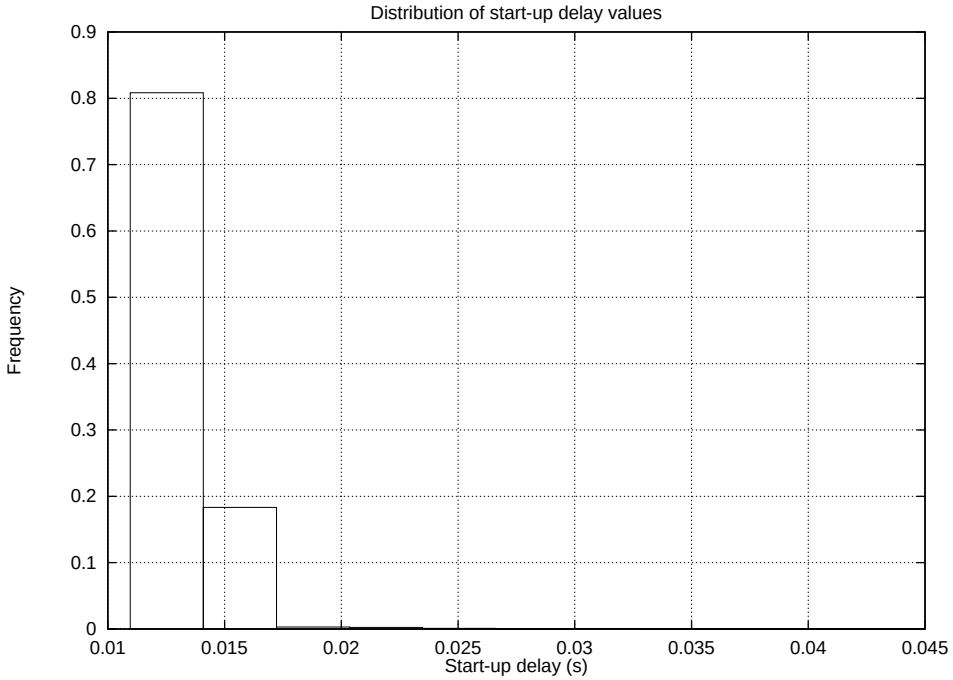


Figure 6.5: Distribution of delay values between the generation of a new connection and the time the traffic generator is started.

Mean	Var	Std. Dev.	95 % CI
8.8048E-03	1.6678E-6	1.2914E-3	3.7556E-05

Table 6.1: Statistical properties of a set of start-up delay values. A start-up delay is the period between the generation of a new connection and the time the traffic generator is started.

6.3.2 Repeatability

In order to reliably compare and contrast different CAC algorithms under a range of connection loads repeated experiments need to give high repeatability of results. Running consecutive experiments with no changes in parameters should reveal as near to identical results as is possible. This section reports on results appraising the repeatability of the CAC test-rig. Firstly, several sets of repeated experiments are contrasted with experiments run with a variety of random seeds. These random seeds form the inputs to the connection arrival and connection holding time distributions; both distributions are random with a negative exponential distribution and each is independent of the other. Seed values also are used in the creation of traffic generators on the physical generator. Each traffic generator uses these random seeds to seed the random number generators that will give distributions of cell burst length and inter burst time (Section 2.2).

The objective was to establish that variations in results for experiments repeated with identical parameters and variations in results for experiments repeated with different random number seeds give a degree of variation in the results. These results would then confirm that the use of one set of random number seeds throughout all experiments would not cause peculiarities in the repeatability of experiments. The second round of repeatability experiments were more complex.

In the second round of repeatability experiments, experiments were run with identical sets of parameters. 100 experiment runs were performed and a statistical evaluation of the accuracy of the repeated results is given. The second round of results used experimental parameters that were near identical to the experiments required to evaluate the CAC test-rig; the objective of this second experiment round was to evaluate the repeatability of the test-rig under conditions as near identical to the evaluation conditions as possible.

The first repeatability tests were made using with connections carrying the TP10S1 source type described in Section 2.2.1. The MCAR of these connections was 10 connections per second and the MCHT was 10 seconds per connection. The CAC algorithm was the simple BT adaptive algorithm for which a threshold of 4.26 was used. This means that new connection attempts were rejected if the link utilisation was above 42.6 Mbps. This value was used because, as documented in Section 3.4, this threshold value was calculated by our BTL collaborators as part of the evaluation. The results obtained for mean line utilisation (MLU) and for the CLR from a buffer of 100 cells in length were compared for each experiment.

Figure 6.6 graphs the values of mean line utilisation from five experiments using the same seed and five experiments with five different seeds for the random number generator. The statistical properties of the two sets of experiments are documented in Table 6.2. It is interesting that the variation between experiment runs occurring in experiments without any change in parameters is slightly smaller but of the same order of magnitude as the variation between experiment runs where the seed of the random number generators has been altered for each experiment run. This implies that the variation in an individual experiment caused by the differences between consecutive runs is almost as large as the 'random' variation of changing the seeding of the random number generators. In addition,

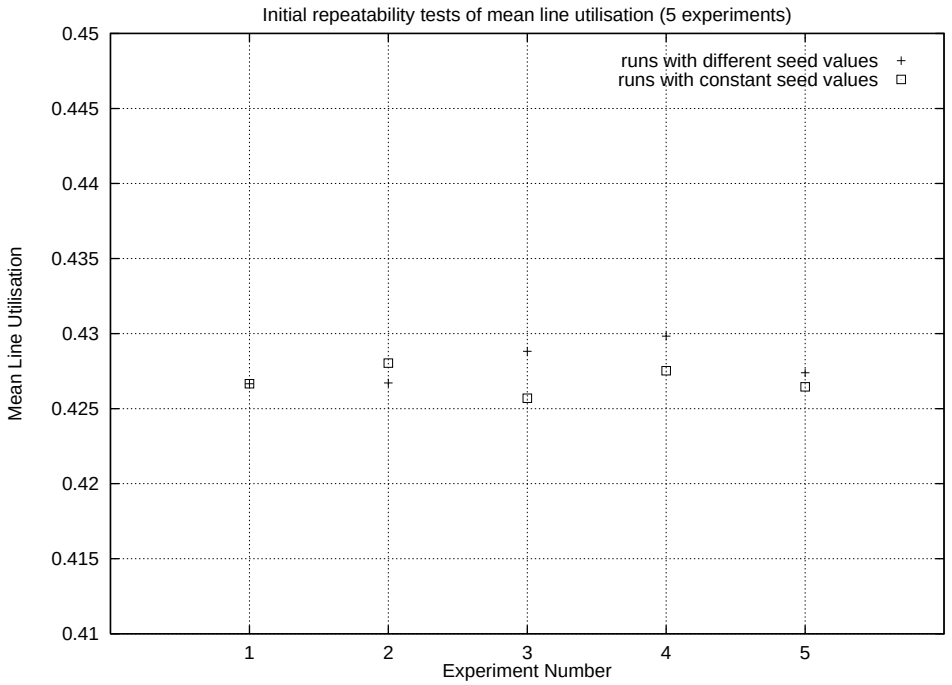


Figure 6.6: Repeatability test results showing mean line utilisation values for 10 repeats of the same experiment.

	Mean	Var	Std. Dev.	95 % CI
varied seed	4.278841E-01	1.935880E-06	1.391359E-03	1.938881E-03
same seeds	4.268741E-01	8.443734E-07	9.188979E-04	1.280499E-03

Table 6.2: Statistics for the values of mean line utilisation for 10 repeats of the same experiment.

these experiments show that, for mean line utilisation at least, the selection of one particular set of random seeds for the generators does not artificially constrain the range results obtainable. The next set of results to compare were the CLR values obtained.

Figure 6.7 shows graphically the values of the CLR (for a buffer length of 100 cells) of the same five experiments with the same set of seed value and the same five experiments with five different sets of seed values. Statistical properties of the CLR experiments are documented in Table 6.3. Once again, similar to the results for mean line utilisation, the results indicate that variation of results for CLR are slightly smaller for experiments repeated with the same set of random seed values than those repeated with different random seed values although having a similar magnitude. Additionally, like the mean line utilisation results, this implies that using one particular set of random numbers will not artificially constrain the variation in results and that the variation is almost as large in consecutive runs for the case where the set of random numbers is kept the same as it is for the case where the set of random numbers is varied.

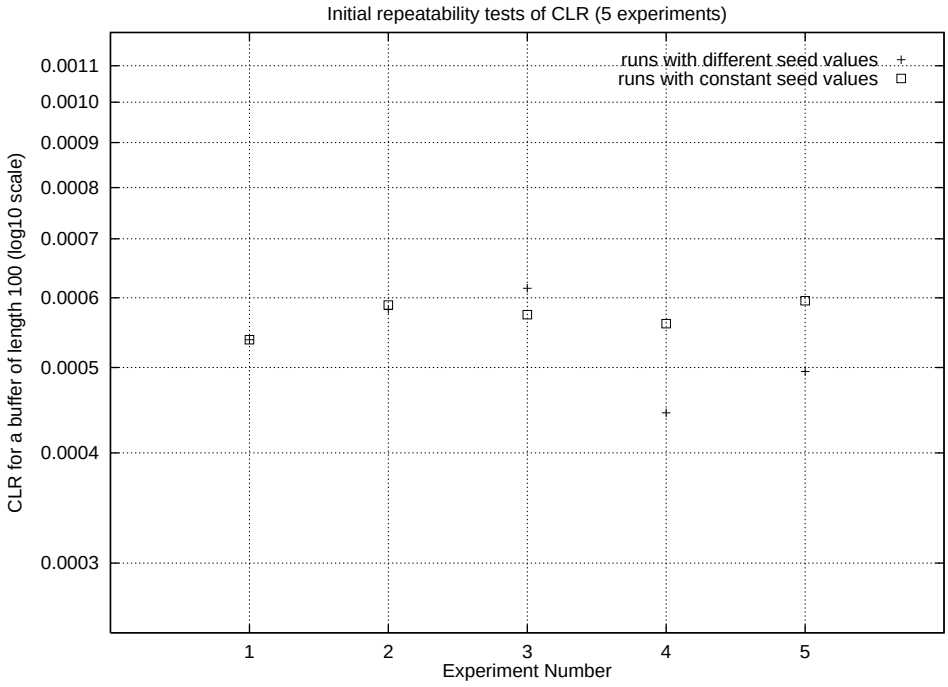


Figure 6.7: Repeatability test results showing CLR values on a 100 cell buffer of 10 repeats for the same experiment.

	Mean	Var	Std. Dev.	95 % CI
varied seed	5.347374E-04	4.618412E-09	6.795890E-05	9.470178E-05
same seed	5.712818E-04	5.356902E-10	2.314498E-05	3.225289E-05

Table 6.3: The statistics for values of CLR of a 100 cell buffer from experiments repeated with and without variations in the set of seed values.

Once the variation between experiments runs without varying any parameters was shown to cause as much variation in the results from experiments as those experiments where the set of random number seeds was changed, the exact variation needed to be established with a larger set of repeat experiments of identical parameters.

In order to establish a more representative and comprehensive sample, 100 complex[6] experiments, involving a mixture of different types of traffic streams, were run. In all possible ways input parameters were held as constants throughout successive experiments on the test-rig.

The experiments themselves involved a mixture of connections of two different traffic types. One of these traffic types was the video source VP10S1 (described in Section 2.3.1) being carried on connections that had an MCAR of 5 connections per second and an MCHT of 10 seconds per connection; the other traffic type was VP5S2 (described in Section 2.3.2) being carried on connections that had an MCAR of 5 connections per second and an MCHT of 5 seconds per connection. This combination of connections and calls gives an offered load of 7.5. The threshold value used in the CAC algorithm was 4.575, this is equivalent to admitting any connection when the line utilisation was below 61Mbps and rejecting new connections when the line utilisation was below this value.

During evaluation of CAC algorithms, the overall CLR and mean line utilisation of an experiment are significant comparison criteria, as a result it was these results that were commonly compared between experiment runs. Figure 6.8(a) shows the mean line utilisation values for the batch of 100 identical experiments. A statistical summary of this collection of results is in Table 6.4 and the distribution of the results is shown in Figure 6.8(b). In comparison, Figure 6.9(a) shows the CLR values for the batch of 100 identical experiments. The statistics of this collection of results is in Table 6.5 and the distribution of the results is shown in Figure 6.9(b).

It is clear that even for experiments with a narrow distribution of mean line utilisation, the values for cell loss ratio have a much greater distribution. This will mean that with a 95 % confidence the link utilisation value will have an error of $\pm 0.21\%$. With a 95 % confidence, the CLR results will give experimental results with an error of $\pm 4.6\%$. Another interesting point from these results is that the link utilisation figures have a distribution that implies the values will be distributed lower than the mean rather than evenly distributed around the mean. Appraisal of the repeatability of the CAC test-rig has given results indicating the boundaries of accuracy for results from the CAC test-rig. There are few avenues for improving on these results without substantial changes to the experiment parameters or the physical equipment in use; however these results appraise repeatability and give a prediction of the accuracy for the CAC test-rig.

It is difficult to decide if the results are satisfactory, this is because evaluation test-rigs for CAC algorithms are not common and not directly comparable. It is worth noting though that simulated or theoretical results can have substantial error margins in the results;

[6]By complex, we mean that these experiments did not involve the use of one traffic source alone, but rather they more closely mimicked experiments run during the later stages of the CAC evaluation. However the experiments run in this set of tests are not comparable themselves to the evaluation experiments, only the repeatability of the CAC test-rig is usefully revealed.

Mean	Var	Std. Dev.	95 % CI
5.477709E-01	3.384366E-05	5.817530E-03	1.160403E-03

Table 6.4: Statistical information on the 100 mean line utilisation results shown in Figure 6.8(a).

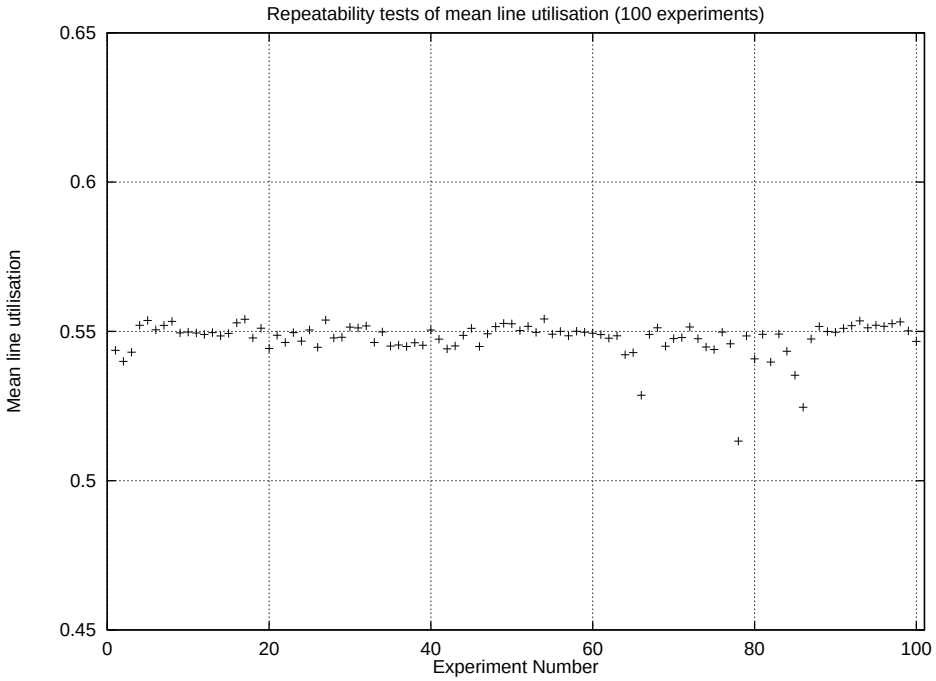
Mean	Var	Std. Dev.	95 % CI
1.238580E-03	9.047702E-08	3.007940E-04	5.969022E-05

Table 6.5: Statistical information on the 100 CLR results shown in Figure 6.9(a).

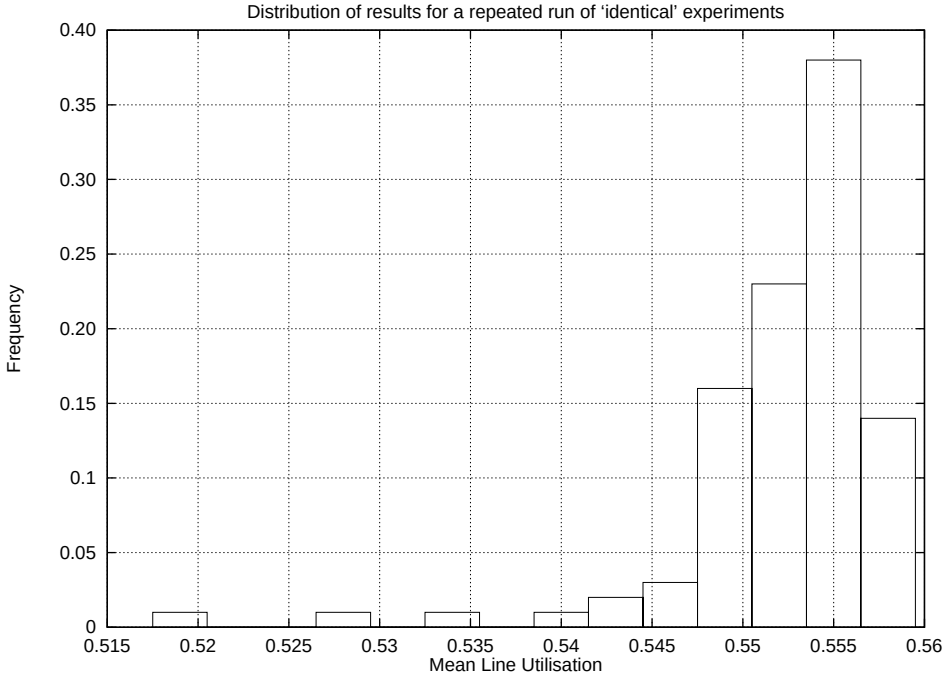
this fact is made clear in Figure 5.9(b) where only the theoretical (simulated) results are documented. In this way the variation in results on the CAC test-rig may be considered no worse than results from a variety of theoretical runs. This concept was further reinforced by the previous sets of 'repeat run' results. Recalling from the comparison between repeated experiments with no parameter changes and repeated experiments with changes of the seed set for the random number generators it was noted the amount of variation in results was the same magnitude for and almost as higher than the repeated experiments where the set of seed values of the random number generators were held constant. As a result it was concluded that documenting the variation was all that should be achieved at this stage.

6.3.3 Adaptability

The final criteria for the CAC test-rig was adaptability. In order to offer the ability to compare and contrast CAC algorithms under a variety of traffic and load conditions in addition to allowing easy implementation of new CAC algorithms, the CAC test-rig design offered a wide variety of new connection regimes. The manner in which new connections will enter the system, combined with the length a successful connection will be held operational can be varied; specified as part of a mathematical model or as trace-driven entries from a pre-prepared log. In addition to a variety of connection regimes, a new physical traffic generator offers the ability to generate a wide variety of traffic types. The traffic could be generated off-line, collected off-line and replayed or created on-line using the desired mathematical characteristics. These aspects together mean a CAC algorithm can be tested under a wide combination of connection load and traffic types. Finally the CAC test-rig must be able to make measurements required for the CAC algorithm and measurements that can be used to appraise the CAC algorithm. The list below records on the available methods for generating new connections, the variety of traffic types, the various measurement methods available and the CAC algorithms currently implemented.

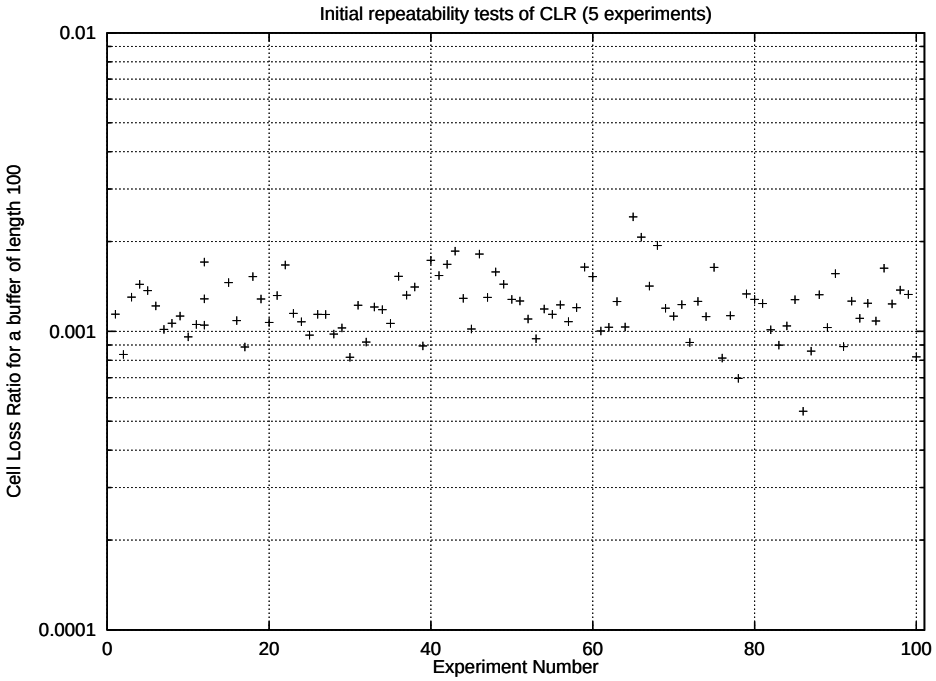


(a) MLU values

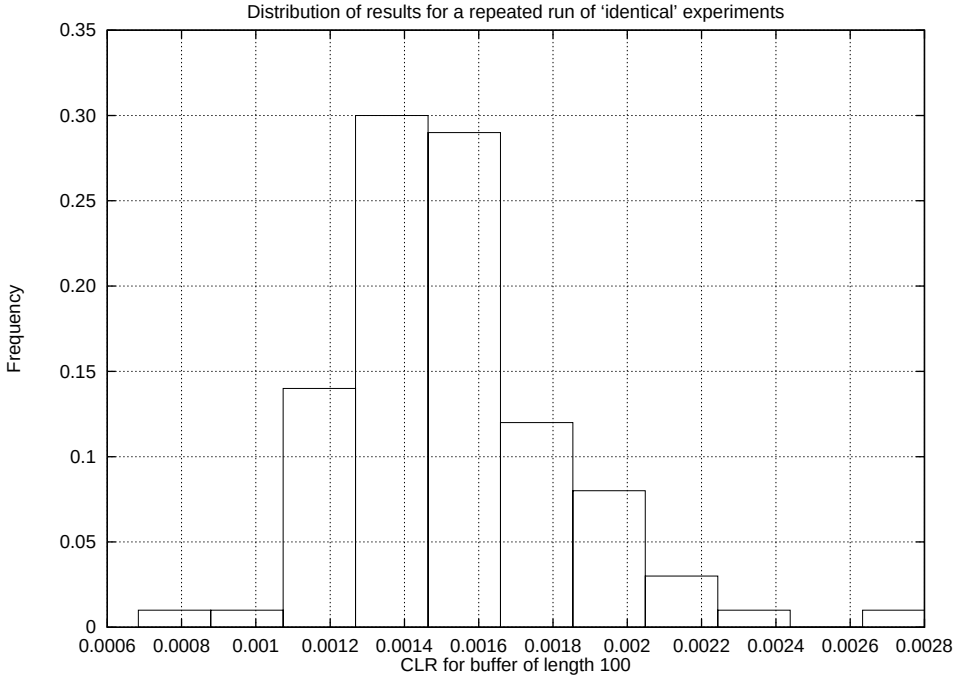


(b) Distribution of MLU values

Figure 6.8: Repeatability test results.



(a) CLR values



(b) Distribution of CLR values

Figure 6.9: Repeatability test results.

Connection Methods

- **New-connection arrival rate described by a mathematical model** This allows a user to specify an MCAR and new connections will arrive with a arrival rate based on a negative exponentially distributed random variable with the nominated MCAR.
- **Connection holding time described by a mathematical model** This allows a user to specify an MCHT and new connections will have a connection holding time based on a negative exponentially distributed random variable with the nominated MCHT.
- **New-connection arrival rates are read from previously created file** This allows logs of systems such as Web servers to be processed into appropriate format logs detailing a connection by connection basis.
- **New-connection holding times are read from previously created file** As in the connection arrival rate case, connection holding times are read from a previously created file.

Traffic types

- **Current**
  - **2-state Markov model** This traffic type, detailed in Section 2.2 transmits cells in bursts. The bursts have a nominated MBS and are transmitted at a nominated PCR. The burst size and inter-burst spacing are negative exponentially distributed random variables with means selected to satisfy the nominated MBS and a nominated value of SCR.
  - **Trace-drive cell list** This traffic type allows traffic traces to be generated off-line and stored in the memory of the physical traffic generator. The trace is created off-line, such a method is highly flexible give the variety of traffic it can represent. This method is used currently for the transmission of traffic representing video sources.
- **Planned**
  - **Hurst traffic model** The incorporation of a theoretical traffic model with an adjustable amount of self similarity (as represented by the Hurst parameters) allows testing against a traffic type that does not have the same multiplexing characteristics as the 2-state Markov model, yet can be easily compared against simulations. Such an implementation of a Hursty source is [29].
  - **Direct video list conversion** This technique is intended to incorporate the off-line generation of video traces into the physical generator itself (Section 2.3). This would be desirable as it allows the much smaller sized lists of video frame

sizes to be kept in memory and converted into cells directly rather than doing this conversion off-line and keeping a full ICT list in memory. This technique is achieved by the addition of various traffic shapers, leaky bucket and chain bucket, to obtain the desired traffic characteristics.

Measurement methods

- **Current**
  - **Line utilisation** This measurement allows the return of line utilisation at the accuracy of one cell down to time periods down to one cell-time.
  - **Cumulative cell count** This measurement is at the accuracy of a single cell over time periods down to one cell-time.
  - **Recent cell count** As 'Cumulative cell count', but returns the count of cells received by the switch since the last 'Recent cell count' enquiry.
  - **Cumulative CLR over total connection** This measurement returns the CLR for the aggregate of cells through the switch.
  - **Recent CLR over total connection** As 'Cumulative CLR over total connection' but returns the CLR for the aggregate of cells through the switch based on the cells received at the switch since the last 'Recent CLR over total connection' enquiry.
  - **Cumulative queue length histogram.** This measure returns the queue length histogram for all cells received by the switch. This information can then be used to create graphs of the Cumulative CDF at the ATM switch buffer.
  - **Recent queue length histogram.** As 'Cumulative queue length histogram.' but returns the queue length histogram based on cells received at the switch since the last 'Recent queue length histogram.' enquiry.
- **Planned**
  - **Cumulative cell count per connection** This measure is planned to return the total number of cells received by the switch for a particular connection.
  - **Recent cell count per connection** As 'Cumulative cell count per connection' but will return the count of cells received by the switch for a particular connection since the last 'Cumulative cell count per connection' enquiry.
  - **Cumulative CLR per connection** This measure is planned to return the CLR for a particular connection based on all cells received by the switch for a particular connection.
  - **Recent CLR per connection** As 'Cumulative CLR per connection' but will return the CLR for a particular connection based on cells received by the switch since the last 'Recent CLR per connection' enquiry.

CAC algorithms

- **Current**
  - **Peak Rate Allocation** This algorithm is the technique specified in [1] by the ATM Forum. It uses no measurements of the line utilisation; instead a worst case scenario is assumed and connection admission is based on the declared PCR of each of the new and current connections in the system being compared with the total line capacity.
  - **BT adaptive CAC algorithm** This algorithm, described in Section 3.4, uses comparison of a known thresholding value with instantaneous line utilisation measurements to allow new connection admission or rejection.
  - **Measure algorithm** The Measure CAC algorithm used is a simplified implementation of the large deviation [15, 16] based mechanism discussed in [13, 12, 14, 4].
- **Planned**
  - **Enhanced BT adaptive CAC algorithms** These algorithms would be threshold based like the CAC algorithm of Section 3.4 however would expand the mechanism to incorporate rudimentary adaptation to mixtures of traffic types. This would allow evaluation of the real-time adaptation of the BT adaptive CAC algorithm [19, 23, 30].
  - **Enhanced Measure algorithm** This family of algorithms expand on the current Measure technique but give unknown, but perhaps significant advantages over the current technique.
  - **CLR algorithms** This algorithm would used the measured CLR from the switch as direct input to the CAC mechanism. Such an algorithm would form a useful comparison as it could potentially give 'perfect' results using a measurement not commonly available to CAC algorithms.
  - **Mean and Variance based algorithm** This algorithm would be based on work by M. Grossglauser [20] and is another approach to measurement based CAC algorithms.
  - **Exponentially weighted utilisation** This algorithm would base descriptions on a recorded history of utilisation, however to avoid temporary level transitions unduly affecting the CAC mechanism, the history would carry a 'weight' in proportion to how recent the utilisation sample had been taken.

Even without taking into account methods as yet unimplemented or those methods only in implementation, it can be seen from this list that the variety of methods and implementations have meant the CAC test-rig well satisfies the criteria of being an adaptable flexible platform for CAC evaluation. As a result of these choices the flexibility has meant

the CAC test-rig offers a satisfactory level of options and choices. In addition to a variety of options for new connection methods, traffic types, measurement methods and CAC algorithms, the CAC test-rig gives a high level of event logging. Such a detailed amount of event logging combined with the wide choice of measurement technology, new-connection regimes and traffic types make the CAC test-rig the ideal platform for CAC evaluation.

7 Evaluation of CAC Algorithms

This section reports on the results gained using the implementation of the BT adaptive CAC algorithm detailed in Section 3.4. Following a discussion of specific implementation details in Section 7.1, results are presented from experiments conducted using the BT adaptive CAC algorithm. In Section 7.2, a first set of experiments indicate the accuracy of the threshold values calculated by our BTL collaborators (given in Section 3.4). Another experiment established the relationship between the thresholding value and the period over which instantaneous utilisation measurement is taken; all other experiments use an instantaneous utilisation measurement period of 30.6ms. The final round of experiments report on the theoretical traffic type, TP10S1, were to empirically establish the relationship between the threshold value, CLR and mean line utilisation. The following sets of results were conducted using alternative traffic types.

Section 7.3 details experiments to empirically establish the relationship between the threshold, CLR and mean line utilisation were conducted for experiments where the connections carried the video traffic type VP10S1. This traffic type has broadly similar characteristics to the theoretical traffic source TP10S1; the same PCR, SCR and MBS. However, this traffic source has a different distribution of burst sizes and a highly periodic structure to the timing of the bursts themselves. The result is an instructive set of comparative results revealing that knowledge of the broad traffic descriptors alone may not reveal enough information to establish the threshold and hence acceptance boundary accurately. The following Section 7.4 reports on experiments to empirically establish the relationship between the acceptance boundary, CLR and mean line utilisation for connections carrying video traffic type VP5S2. These results along with those of Section 7.3 form an interesting comparison with the results of the next Section.

The next section, Section 7.5, gives results establishing the relationship between the acceptance boundary, CLR and mean line utilisation for several experiments were the connections entering the CAC test-rig are carrying a mixture of traffic types. The mixture of traffic types carried by the connections of these experiments are a sliding ratio of the two traffic sources VP10S1 and VP5S2. Based on these results, a relationship between the traffic mix, acceptance boundary, CLR and mean line utilisation is established.

Section 7.6 compares the results gained using the BT adaptive CAC algorithm with results gained using two other CAC algorithms. The alternative CAC algorithm used are the Peak Rate allocation policy and the Measure algorithm. The Peak Rate allocation policy assumes a worst case scenario about new connections and new connection admission is based on the declared PCR of each of the new and current connections in the system being compared with the total line capacity. The Measure algorithm is a naïve implementation of the large deviation [15, 16] based mechanism discussed in [13, 12, 14, 4]; its actual implementation is discussed further in Section 7.6.

The results of Section 7 required over 1,100 separate experiments, each experiment ran for approximately 2 hours. Each experiment carried at least 1×10^8 cells, this value, established in Section 5, came as a trade off between experiment run-time and the accuracy of results, such as the CLR value for an experiment. Increasing the number of cells sent

through the system by a magnitude would also increase the running time by a magnitude, for a handful of experiments such an increase may be manageable, however for over 1,100 experiments, such an increase of a magnitude would not have been acceptable.

7.1 Implementation of BT adaptive CAC algorithm

The BT adaptive CAC algorithm relies on instantaneous measurements of the line utilisation. An instantaneous measurement of the line utilisation is required as part of each connection admission request. The instantaneous measurement is compared against a thresholding value the CAC is currently using and if the measured line utilisation is equal to or less than the thresholding value the CAC can admit the new connection request. The measurement of instantaneous line utilisation is critical to the operation of a threshold-based CAC algorithm like the BT adaptive CAC algorithm.

As stated in Section 3.5 an implicit assumption throughout the theory of the BT adaptive CAC algorithm is that instantaneous measurements of traffic can be made for each and every new connection request on which the CAC algorithm must decide. The period of an instantaneous measurement is not zero and the range of values include the time during which other new connections will request admission into the system

This means that there could be a need for many measurements taken simultaneously while new connections are being decided on by the CAC algorithm. The hardware of an ATM switch is not able to perform such simultaneous instantaneous measurements with the desired period. However periodic measurements of this resolution and measurement period can be realised.

Periodic measurements are ones where the switch routinely returns counts of cells traversing the switch. These cell counts give the line utilisation for each measured period for consecutive periods of time. When a line utilisation measurement is requested, the measurement for the most recently completed measurement period is returned. Using Periodic utilisation values as a substitute for instantaneous line utilisation measurements will incur an error: the difference between the line utilisation taken over the period and the line utilisation as it would have been had the measurement been taken instantaneously.

An error in the measurement period may result in an error in the line utilisation figure. This error arises because the true line utilisation may have been dramatically different from the measured value. The situation is shown in Figure 7.1; a new connection requires a measurement of the utilisation, the instantaneous utilisation would give a value of 25% while the most recent periodic measure gives 50% utilisation. In this example the periodic measurement of utilisation differs from the line utilisation measurement by 25% of the total line rate. The error will display a roughly normal distribution with minimum and maximum values being the minimum and maximum values of utilisation measurable in any single measurement period. How this will actually affect the algorithm is less well understood; an assumption has been that the long term effect will be negligible. The conclusion about this effect being negligible is because the system has an equal likelihood of admitting a connection attempt as it has of rejecting a new connection attempt. It is worth reiterating that a real switch cannot make the many simultaneous measurements of

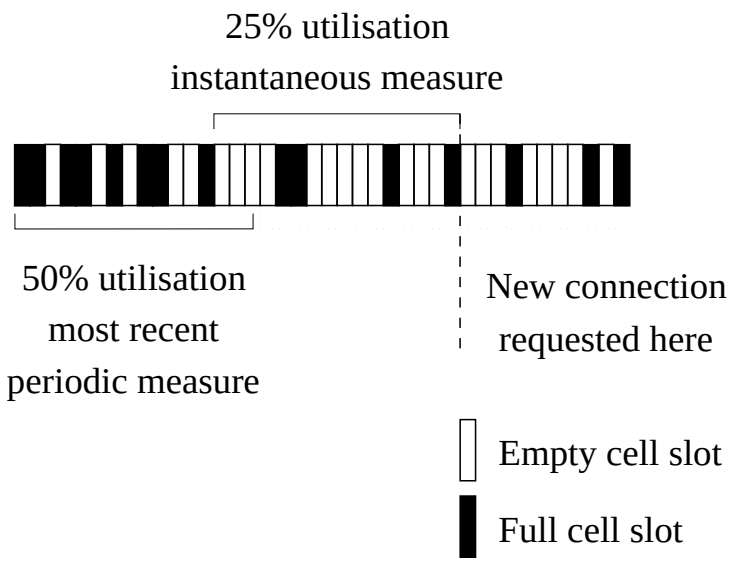


Figure 7.1: Instantaneous and period measurements of line utilisation.

utilisation the BT adaptive CAC algorithm assumes are available; after consulting with our colleagues at BTL the periodic measurement technique discussed here was the best implementable alternative.

Once implemented, the logging system in the CAC test-kit gives sufficient information that real-time graphs can be produced displaying information: on the current line utilisation, the current connections in progress and information about new connection acceptance or rejection. Figure 7.2 shows the plot of 100 seconds of time from the start of an experiment. The figure consists of three graphs stacked vertically. The x axis in all cases is time, shown in seconds since the start of the experiment. The top graph in each set of 3 produced by the on-line system shows the connection arrival process. For each connection which arrives a vertical bar is drawn. In the event that an arriving connection was accepted by the CAC algorithm, a green vertical bar is drawn. If the connection is rejected, then the vertical bar is red, and extends downwards. In the leftmost part of Figure 7.2, where the time is less than 45 seconds, no new connections have been rejected because measurements of the instantaneous line utilisation are less than or equal to the thresholding value. Whereas, after 45 seconds, sufficient traffic is now in the system for the instantaneous line utilisation to be above the thresholding value and as a result for connections to be rejected.

The second graph from the top is a display of the traffic dynamics in the switch, measured in real time. The thresholding value in use by the CAC algorithm is shown along-with the current measure of instantaneous line utilisation. The bottom graph of Figures 7.2 and 7.3 shows the number of connections in progress in the system, over

time. This rapidly climbs, as new connections enter the system at a greater rate than they clear down because in the empty system (at time zero) no connections are rejected. Once rejections occur, the number of connections in progress stabilises, but displays the expected variation due to statistical fluctuations. In this experiment connections carried the theoretical traffic type TP10S1 discussed in Section 2.2.1, an acceptance boundary of 4.26 and thus an thresholding value of 42.6 Mbps was used, the MCAR was 10 connections per second with an MCHT of 10 seconds.

When the system is started there is no load in the system as a result new connections can be admitted into the system. Box 1 shows where measurements of current utilisation are below the threshold value and as a result new connections are admitted by the CAC algorithm. In comparison, Box 2 shows where measurements of the utilisation figure are above the threshold value and as a result new connections are rejected by the CAC algorithm. Beyond the 60 second mark it can be seen that the number of connections in progress stabilises as does the current utilisation.

Figure 7.3 shows results from the CAC test-kit when an experiment using the BT adaptive CAC algorithm has been running for a substantial period of time. The stabilisation of the number of connections in progress and the current utilisation figure are well demonstrated with the system running after a substantial period of time. It is worth recalling Figure 3.1 from Section 3 to appreciate the decision process the CAC algorithm is using as new connections attempt admission. The CAC checks the current utilisation figure and if this figure is less than or equal to the threshold, the new connection is admitted; if the current utilisation is above the threshold, the new connection is rejected.

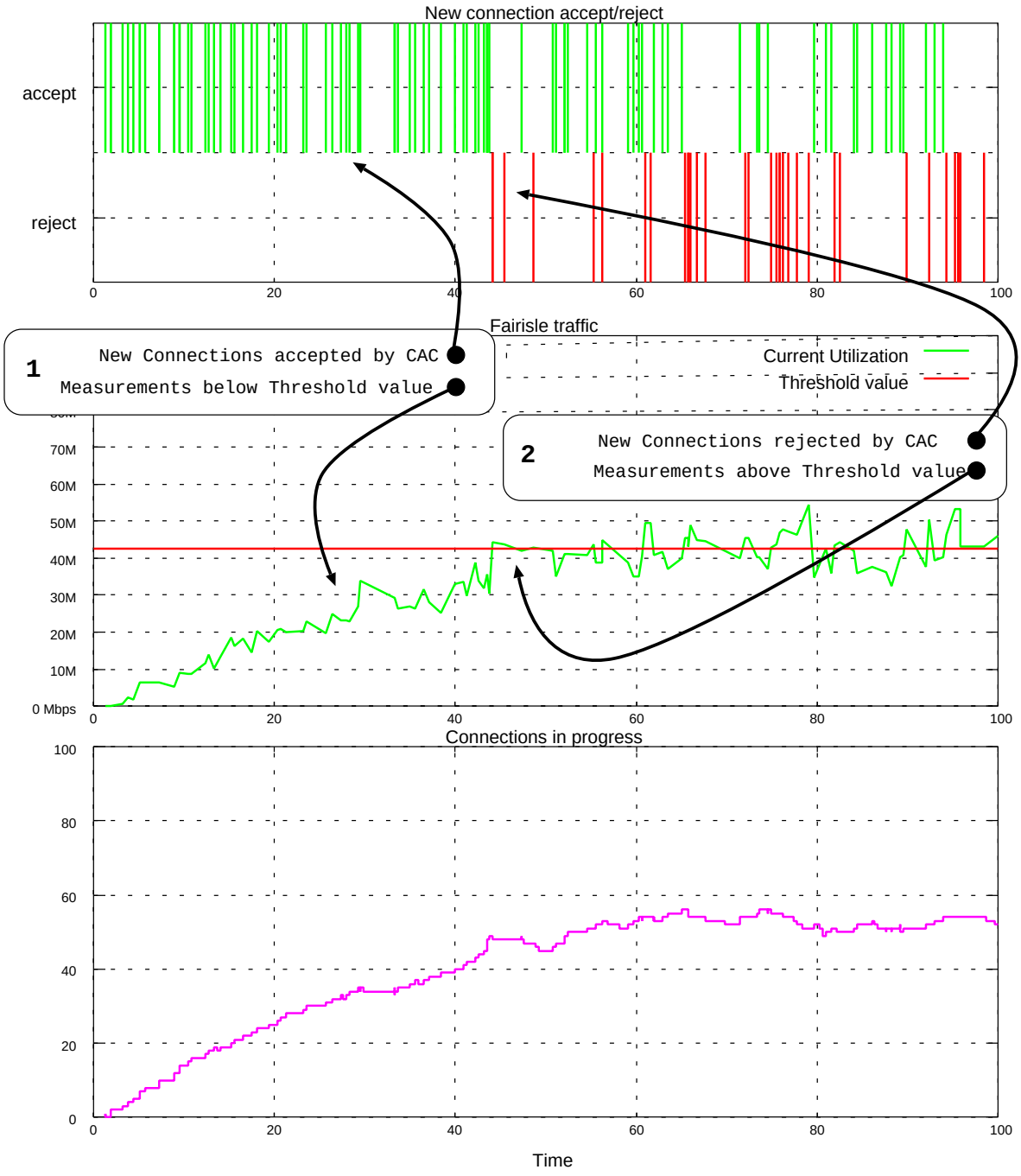


Figure 7.2: The first 100 seconds of operation of a CAC test-kit experiment using with the BT adaptive CAC algorithm.

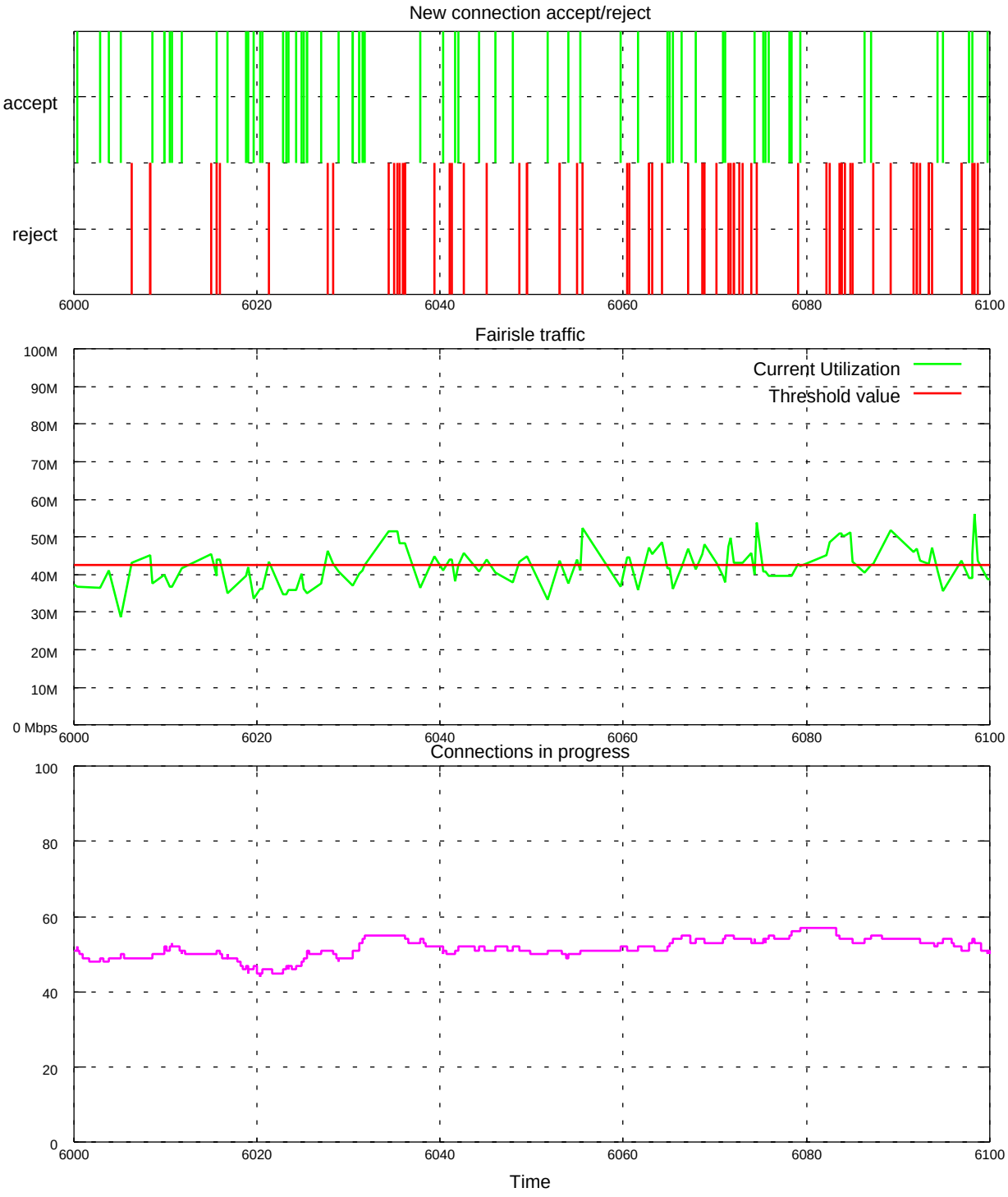


Figure 7.3: 100 seconds during the operation of a CAC test-kit experiment used with the BT adaptive CAC algorithm.

Offered Load	10	4	3
MCAR (per second)	10	4	3
MCHT (second)	10	10	10
To achieve a CLR of 10^{-3}			
Acceptance Boundary	4.26	5.74	7.46
Threshold (Mbps)	42.6	57.4	74.6

Table 7.1: Acceptance boundaries supplied by BT for evaluation.

7.2 Experiments with theoretical traffic type TP10S1

Experiments were conducted with connections carrying traffic type TP10S1. This theoretical source has a PCR of 10Mbps, SCR of 1Mbps and MBS of 25 cells; more details of the source characteristics are given in Section 2.2.1. The connections had a holding time distribution that was exponential with an MCHT of 10 seconds per connection. The connection arrival rate also had an exponential distribution, but with varied values of MCAR so as to achieve varied values of offered load into the CAC system. Table 7.1 gives the acceptance boundary values and the threshold values used for the first stage of experiments. These acceptance boundaries were calculated to give a CLR of 10^{-3} for a buffer size of 100 cells.

The first stage of results was to use the threshold values given in Table 7.1 and compare the results obtained with the parameters used in the creation of the threshold values. One outcome of these first results was that experiments with offered loads of 3 and 4 were not carried on any further. The next stage of experiments with the TP10S1 traffic type shows the changes in CLR and mean line utilisation for a variety of period lengths where the period length is the time over-which the instantaneous line utilisation, an input to the CAC algorithm, is measured. This section confirms the theory of Section 3.5, reinforcing the prediction on upper and lower limits on the length to the measurement period. The final experiments with the TP10S1 traffic type involve establishing the relationship between CLR, mean line utilisation and the threshold value used. One result of this final experiment set is to establish empirically the optimum threshold value needed to give a CLR of 10^{-3} for a buffer size of 100 cells.

7.2.1 Generated threshold values results

Table 7.1 gives the acceptance boundary and thus the threshold value required for three experiments using the theoretically derived traffic type TP10S1. In each of these three experiments the threshold is designed to deliver a CLR of 10^{-3} for a buffer size of 100 cells. The new connections have an exponentially distributed connection holding time with an MCHT of 10 seconds per connection. The new connections also have an exponentially distributed connection arrival rate; the MCAR is dependent on the offered loads. Table 7.1 gives the values of offered load and MCAR in each of the three cases.

Table 7.2 summarises the resulting mean line utilisation, probability of the queue exceeding a length of 100 cells, and the projected CLR value for a buffer length of 100 cells

Offered Load	Acceptance Boundary	Mean Line Utilisation	$P(Q > b)$ for a 100 cell buffer	CLR for a 100 cell buffer
10	4.26	0.4225	5.21×10^{-4}	1.23×10^{-3}
4	5.74	0.3074	6.29×10^{-5}	2.05×10^{-4}
3	7.46	0.2397	2.52×10^{-7}	1.05×10^{-6}

Table 7.2: Results obtained using calculated thresholds for given offered loads.

as a result of this. From these results the BT adaptive CAC algorithm appears pessimistic in its admission policy, allowing fewer connections into the system than could be tolerated for the desired CLR value. This result predicts that if the BT adaptive CAC algorithm is pessimistic it may be wasting line capacity while achieving a given CLR.

The experiments conducted with offered loads other than 10 were not taken further than this round. This decision was taken on the basis that the mean line utilisation could not be raised substantially and the CLR derived from this experiment would not be raised substantially either. The reason for this becomes clearer in Table 7.3.

Table 7.3 shows that the CAC decision rejected no new connections in the case where the offered load was three, for an offered load of four only one new connection was rejected, while for an offered load of ten the CAC algorithm rejected 47% of new connection attempts. The only way to get more statistical significance in the number of connections rejected for the lower values of offered load would be to increase the time over which the experiment is run. As for the case of an offered load of 3 the experiment run time was already in excess of 4 hours and an increase would have to be potentially two (or more) orders of magnitude to create any significant loss, such an approach was not seen as satisfactory. As a direct result, experiments from this point on used an offered load of 10 only.

7.2.2 Effects of varied measurement periods

The period over which the instantaneous measurement of line utilisation is made is a base parameter for the BT adaptive CAC algorithm. Section 3.5 discusses how the period over which the instantaneous measure of line utilisation is made must be in the range $b \times C_t < t_m < H$ Where t_m is the time over which the measurement is to be made, b is the size of the buffer, C_t is the transmission time of a single cell and H is the MCHT. For the experiments of this section t_m, the time over which the measurement is made, needs to be in the range $440\mu s < t_m < 10s$.

The purpose of this section was to take results of the system for a range of values of t_m. Results were taken for values of t_m inside the range established by theory and also results were taken outside this range. The acceptance boundary was a constant 4.26 throughout this set of experiments; this results in a constant threshold of 42.6 Mbps being used throughout. The traffic type carried by the connections was the TP10S1 traffic source of Section 2.2.1. New connections had an exponential distribution of arrival times with an MCAR of 10 connections per second and an exponential distribution of connection holding

Offered load	3	4	10
MCAR (s^{-1})	3	4	10
MCHT (s)	10	10	10
Acceptance boundary	7.46	5.74	4.26
Threshold values (Mbps)	74.6	57.4	42.6
Running time (s)	1535.98	1181.56	866.48
Total cell count	103278532	102275295	102769787
Total connection attempts	4613	4717	8617
Accepted connection count	4613	4716	4570
Rejected connection count	0	1	4047
Connection accept ratio	1.00	1.00	0.53
Mean connections in progress	30.22	39.80	52.96
Mean line utilisation	0.2397	0.3074	0.4225
$P(Q > b)$ for 100 cell buffer	2.52×10^{-7}	6.29×10^{-5}	5.21×10^{-4}
CLR for a 100 cell buffer	1.05×10^{-6}	2.05×10^{-4}	1.23×10^{-3}

Table 7.3: Detailed results of pre-calculated thresholds for given connection loads.

times with an MCHT of 10 seconds per connection. Results show the range of values of t_m versus the link CLR, the mean line utilisation and the mean number of connections in progress.

Figure 7.4 shows the resulting CLR versus the values of t_m: the time period over which the instantaneous utilisation measurement is made. In this figure each point represents one experimental result; several experiments were run for each value of the measurement period. From this graph it is clear that as the period over which the instantaneous measurement is taken approaches 10 seconds the MCHT of the incoming connection requests, the CLR value deteriorates. Such a result can be predicted by the knowledge that as the measurement period is increased sharp, short-term transients in the line utilisation will not be reflected in the line utilisation measurement. Because sharp, short-term transients in the line utilisation will not be reflected in the line utilisation measurement, connections will be more likely admitted when the traffic of those connections cannot be supported at the desired CLR rate. As the measurement period approaches 440μs, the buffer length multiplied by the transmission time of a single cell, the lower value of the range for t_m, the CLR is holding steady or starting to give lower CLR values. Lower values of CLR are predicted because the measurements will not account for the effective length of the buffer itself. Results graphed in Figure 7.4 do not indicate clearly the trend as the value of t_m goes past the lower bound, although lower values of CLR are suggested.

The results graphed in Figure 7.5 show a trend that resembles Figure 7.4. As the period over which the instantaneous utilisation measurement is made is increased the mean number of connections in progress also increases. When this measurement period is

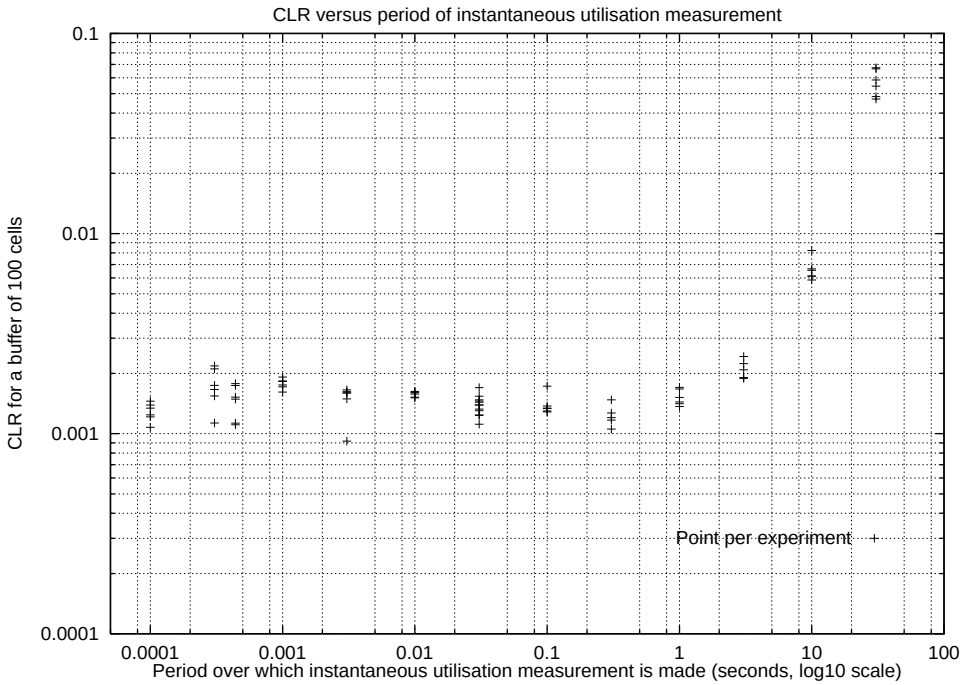


Figure 7.4: CLR versus period of instantaneous utilisation measurement.

decreased, a reduced mean number of connections in progress is suggested. Such a trend correlates well with the understood behaviour of the threshold-based CAC algorithm: as the measurement period is reduced the algorithm will respond more quickly to changes in the level of line utilisation thus the algorithm will maintain the CLR rate in exchange for fewer connections in progress. Larger values of the measurement period will result in a CAC algorithm that is less responsive to shifts in the line utilisation level. The result of this will be that new connections will more often be allowed into the system, when the level was possibly too high, giving higher CLR rates and higher values for the mean connections in progress. Compared with the mean connections in progress: Figure 7.5 and the CLR: Figure 7.4, the mean line utilisation versus the measurement period, Figure 7.6 shows results that are not as intuitive.

Figure 7.6 shows the resulting mean line utilisation from a range of values of the time period over which the instantaneous utilisation measurement is made. This figure shows that as the utilisation measurement period approaches the MCHT of the incoming connection requests, the amount of line activity in the system drops. This result is counter-intuitive, a decreasing mean line utilisation would be thought to give a decreasing CLR for the line. This graph shows that as the measurement period is increased, the mean line utilisation decreases; in comparison, as the measurement period is increased the CLR for the line also increases. The reasons for this may be explained by Figure 7.7.

Figure 7.7 shows the relative frequency distribution of the connections in progress; distributions for several

values of the period of instantaneous utilisation measurement are shown. The medium

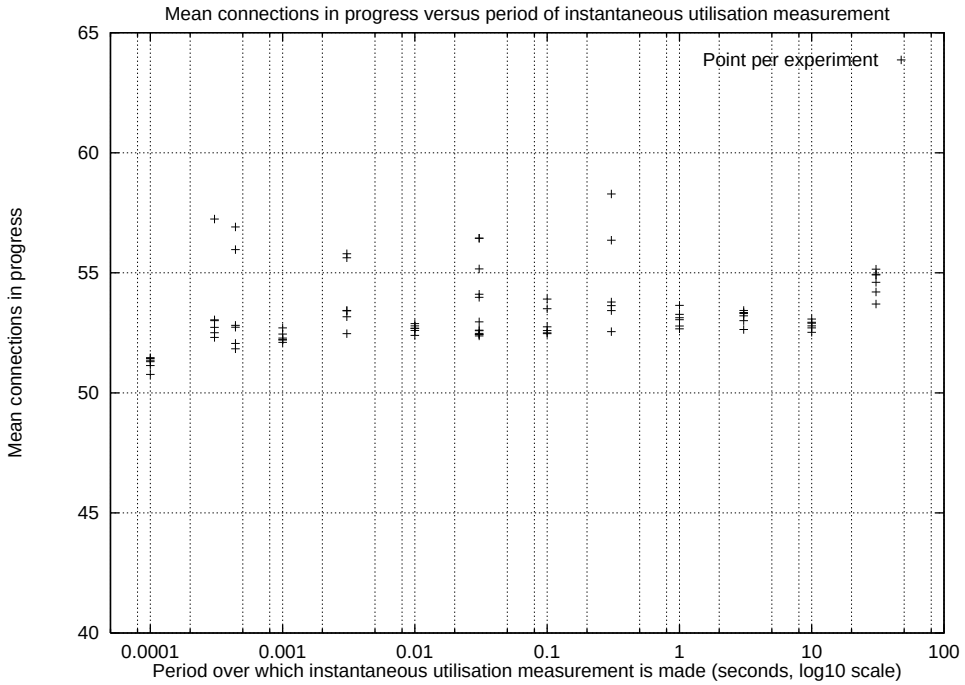


Figure 7.5: Mean connections in progress versus period of instantaneous utilisation measurement.

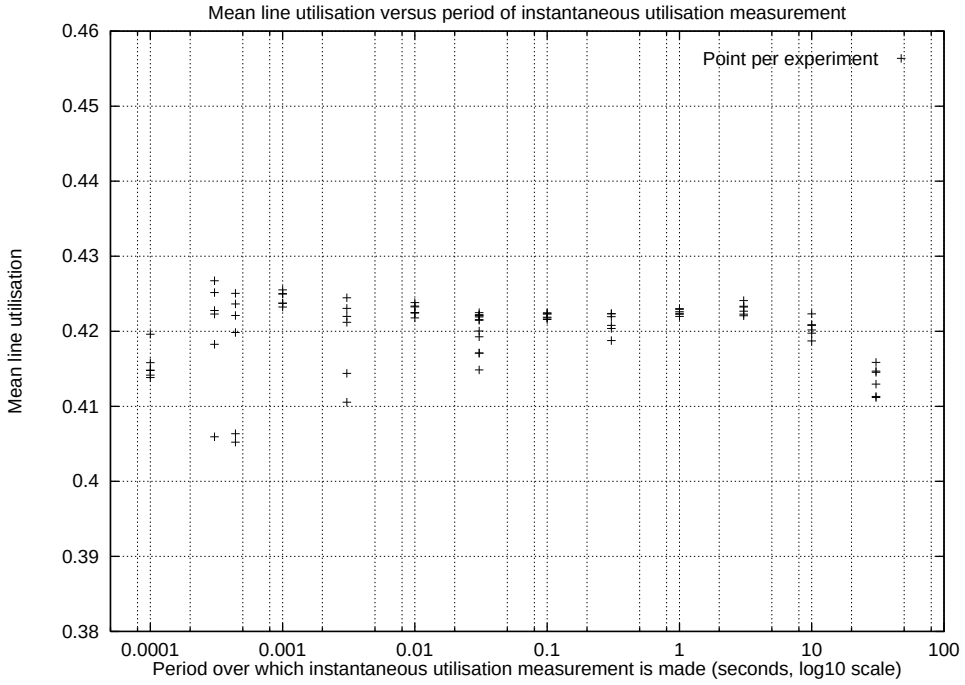


Figure 7.6: Mean line utilisation versus period of instantaneous utilisation measurement.

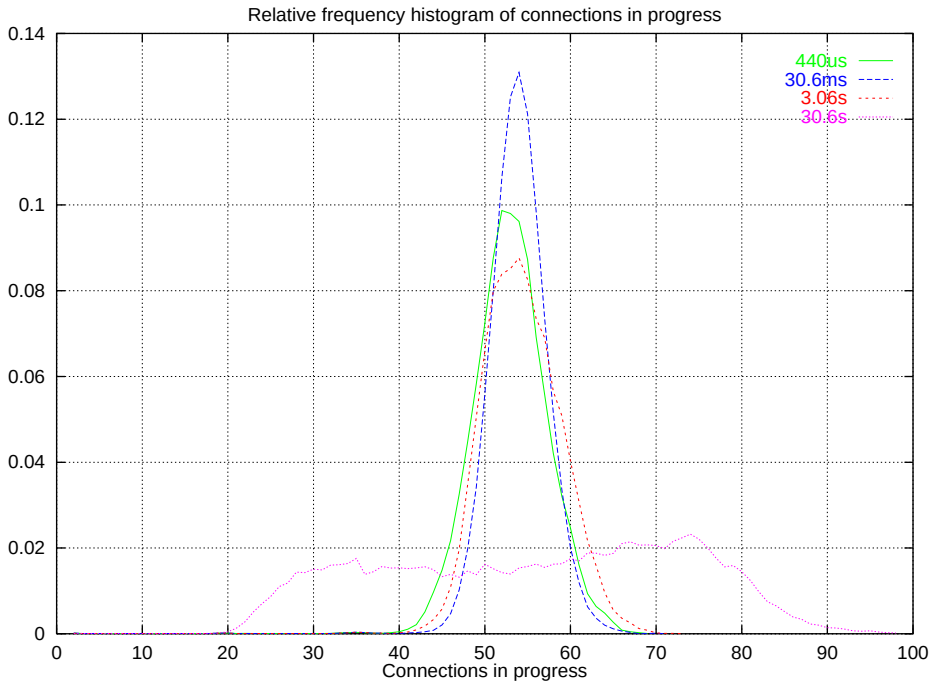


Figure 7.7: Relative frequency distribution of the connections in progress. Distributions for several period of instantaneous utilisation measurement values are shown.

value of 30.6ms gives a distribution with a very small variance value. As the measurement value is increased (3.06s) or decreased (440μs), the variance of the number of connections in progress is increased. When a long period is used, a period that is greater than the MCHT of the new connections, the variance increases significantly for the distribution of connections in progress. These results indicate that large periods of the measurement interval are causing an unresponsive CAC algorithm, a CAC algorithm that the theory of Section 3.5 predicts will not account for all short duration connections in the utilisation measurement. However the distribution for a period of 30.6s shown in Figure 7.7 is not completely symmetrical about the mean (54.757 connections in progress). The result of this may be that connections with a lower mean line utilisation are able admitted more frequently; or that these connections, once admitted, are having reduced effect on the CAC measurements and hence the CAC process.

7.2.3 Empirical threshold results

Section 7.2.1 gave results for the CLR and mean line utilisation achieved with the acceptance boundary, and hence threshold value, calculated by our BTL collaborators. The objective of this section is to develop the relationship with the threshold value and the CLR or mean line utilisation. These results should give the relationship between the acceptance boundary, the CLR and the mean line utilisation.

This section combines results gained using a wide range of threshold values with a

lest-square curve fitting method to give an empirically derived relationships between CLR, mean line utilisation and threshold value. Using this empirical derivation, the value of threshold required to achieve a given target CLR can be calculated directly and this value compared with the entries of Table 7.1.

Figure 7.8 graphs the results of experiments using the TP10S1 traffic type for connections with an MCHT of 10 seconds per connection and MCAR of 10 connections per second. The desired CLR is 10^{-3} for a buffer size of 100 cells; the predicted value for this CLR is shown on the graph in Figure 7.8. Predicted threshold value, 41.19 Mbps, is derived from the equation of the fitted curve. The equation of this curve is given in Table 7.4. As an indication of the accuracy of the fit, the squared correlation is listed alongside the equation of the curve. The fitted curve shows a linear relationship between the threshold and the log of the CLR.

The CLR results of experiments are graphed versus the mean line utilisation results in Figure 7.9. For the threshold-based CAC algorithm a direct relationship can be formed between the mean line utilisation and the CLR; experiments with higher CLR figures also having higher line utilisation. Figure 7.9 also shows a curve fitted to the results of the experiments graphed in this figure. The equation of the curve is given in Table 7.4, alongside a value of mean line utilisation that was comparable to a CLR of 10^{-3} for a buffer size of 100 cells. The fitted curve shows a linear relationship between the mean line utilisation and the log of the CLR.

The final figure of this section, Figure 7.10, shows the threshold values used in each experiment plotted against the resulting mean line utilisation. The results indicate a linear relationship between the threshold value and the mean line utilisation. Figure 7.10 also shows a curve fitted to the results, the equation of this curve is given in Table 7.4. The threshold values used in Figure 7.10, Figure 7.8 and in Table 7.4 are the actual value in Mbps that the threshold-based CAC algorithm will use. The value predicted by theory is the acceptance boundary and as mentioned in Section 3.4, the acceptance boundary is related to the actual threshold used by a value called "link capacity", or simply capacity. The capacity depends on the current mix of traffic types, as a result this value varies for different types of traffic but is constant for situations where the mixture of traffic types remains unchanged. The capacity is calculated by dividing the link rate (100Mbps) by the PCR of the traffic type (10Mbps); the capacity is 10. This simple relationship makes easy the task of converting the fitted lines of Table 7.4 into the relationships of Table 7.5. Table 7.5 does not give a relationship between CLR and mean line utilisation as this is unaffected by the conversion from threshold value to acceptance boundary.

	Best fit equation	Squared correlation	Solution for a 10^{-3} CLR
Threshold in terms of CLR	threshold = 72.2749+ 4.49051 × log(CLR)	0.988	41.26
Mean Line Utilisation in terms of CLR	mlu = 0.6698+ 0.03764 × log(CLR)	0.988	0.410
Mean Line Utilisation versus Threshold	mlu = 0.0645+ 0.00837× threshold	0.998	-

Table 7.4: For TP10S1, equations of the lines of best fit in each of the three relations. Solutions for a CLR of 10^{-3} are listed where applicable.

	Best fit equation	Squared correlation	Solution for a 10^{-3} CLR
Acceptance Boundary in terms of CLR	boundary = 7.22749+ 0.449051 × log(CLR)	0.988	4.126
Mean Line Utilisation versus acceptance boundary	mlu = 0.0645+ 0.0837× boundary	0.998	-

Table 7.5: Acceptance boundary relationships derived from Table 7.4. A solution for a CLR of 10^{-3} is listed for the CLR versus acceptance boundary relationship.

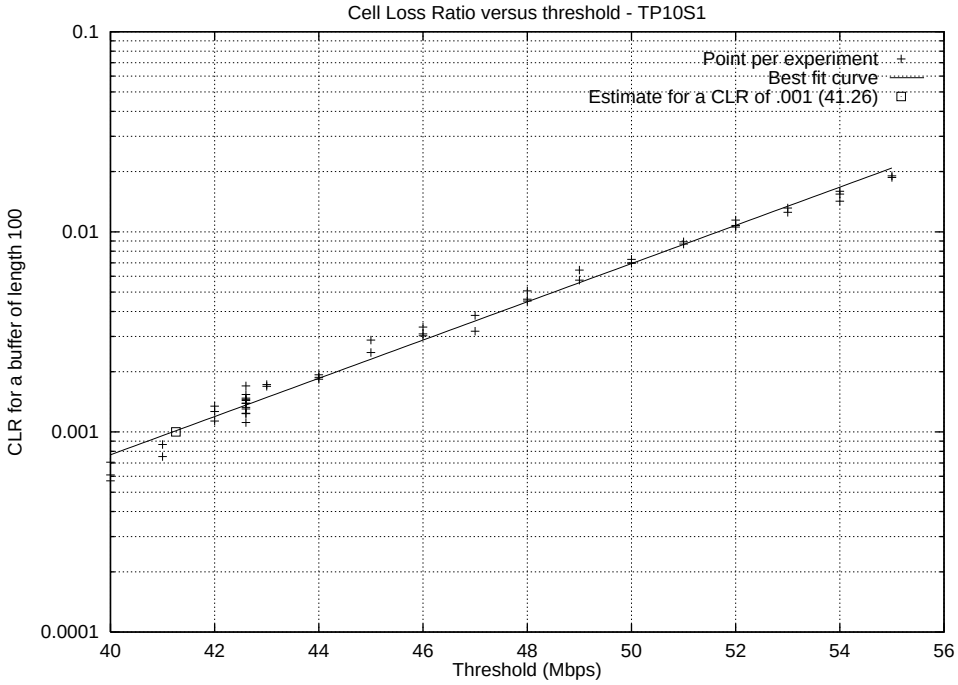


Figure 7.8: CLR versus threshold for TP10S1 experiments.

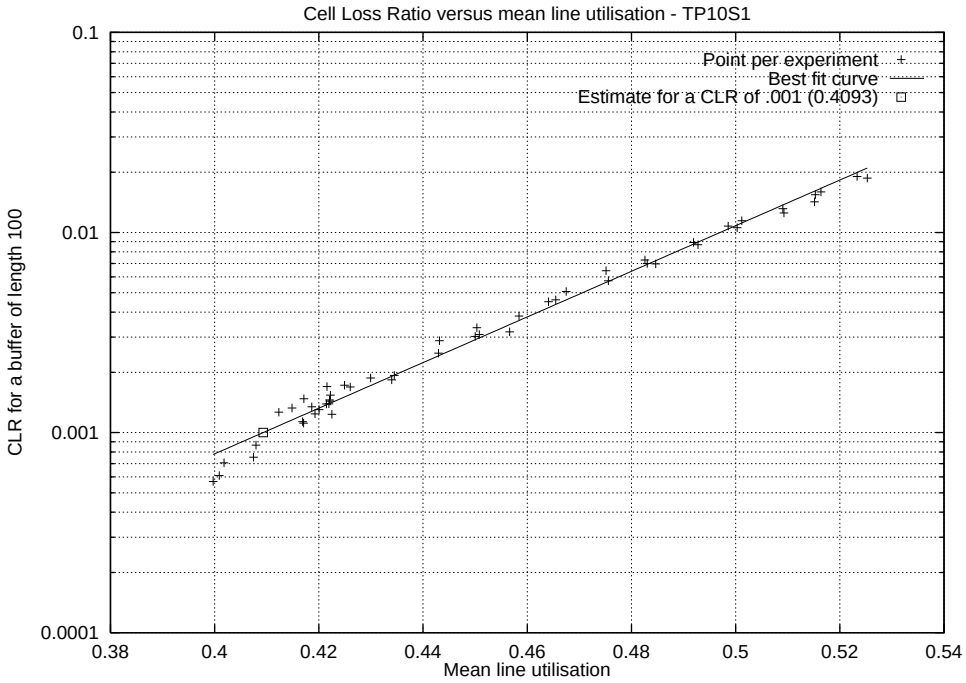


Figure 7.9: CLR versus mean line utilisation for TP10S1 experiments.

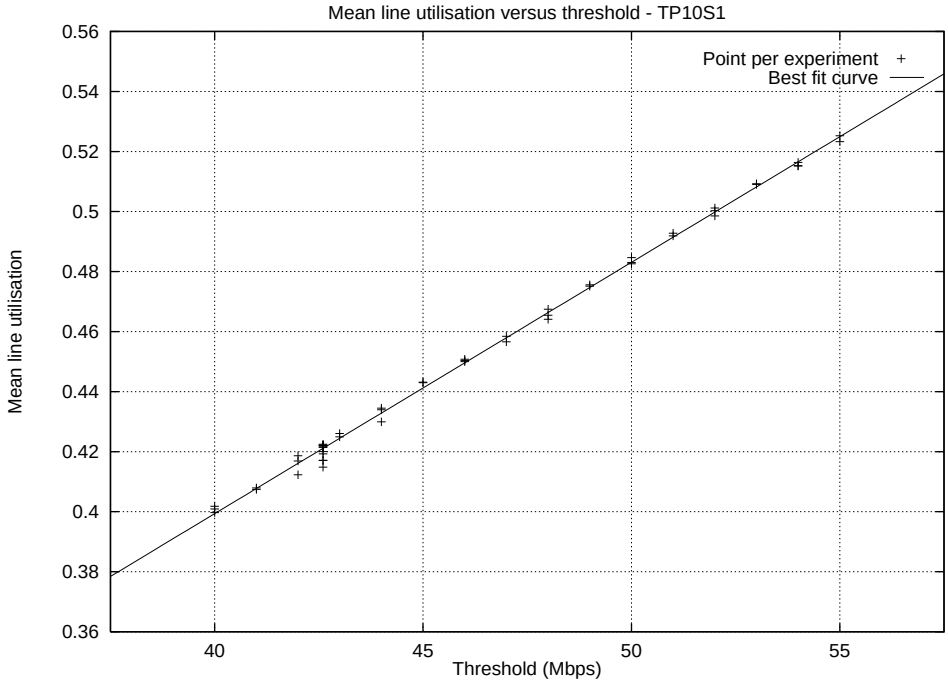


Figure 7.10: Mean line utilisation versus threshold value for TP10S1 experiments.

7.3 Experiments with video traffic type VP10S1

This section, like Section 7.2.3, establishes the relationships between the three variables CLR, the threshold value (and thus acceptance boundary) and mean line utilisation using empirical methods. The results of a series of experiments are graphed and a curve is fitted to the results; thus relationships can be derived for CLR versus threshold, CLR versus mean line utilisation and mean line utilisation versus threshold. The traffic source VP10S1 (detailed in Section 2.3.1) forms an interesting comparison with TP10S1; both traffic types have the same PCR, SCR and MBS however the structure at the burst level is different between these two sources. In particular VP10S1, being based upon a video stream, has a periodic burst structure not present in TP10S1.

The effective load of experiments made using the VP10S1 is the same as experiments made using the TP10S1 traffic type of Section 7.2.3. Both traffic sources have a PCR of 10Mbps and an SCR of 1Mbps (giving an activity of 0.1); and both sets of experiments used an MCAR of 10 connections per second and an MCHT of 10 seconds per connection. This results in an offered load of 10 for both the experiments of Section 7.2.3 and this section.

Figure 7.11 graphs the resulting CLR values for experiments conducted with particular threshold values. This figure also shows a curve fitted to the CLR and threshold results. The equation of this curve is given in Table 7.6; alongside the equation the value of the fits' squared correlation is also given. There is a significantly greater spread in the results of CLR for the VP10S1 traffic type as compared with the results of CLR for the TP10S1 traffic type; the squared correlation is 0.900 for VP10S1 traffic as compared with 0.988 for the TP10S1 traffic. The reasons for such a spread of results is not preciously known however the periodic burst structure of VP10S1 seems likely to have played a significant role. The VP10S1 results have a similar spread of results for CLR versus mean line utilisation.

Figure 7.12 graphs the results of CLR versus mean line utilisation and the illustrated curve is fitted to these results. The equation for the fitted curve is shown in Table 7.6. The spread in this figure is greater than the spread of the CLR versus mean line utilisation for TP10S1; the squared correlation for the VP10S1 traffic is 0.902 compared with 0.988 for the TP10S1 traffic experiments. Once again the wide distribution of results could be due to the periodic characteristics of the VP10S1 traffic.

The periodic nature of a traffic source may cause correlation effects between multiplexed streams of the same traffic type when numerous connections are in progress. When periodic traffic streams become synchronised, if the cell bursts overlap the CLR may potentially be higher while if the bursts do not overlap, the throughput will be as high but with a potentially lower CLR. Such potential correlation of streams could result in the wide variation in results for CLR versus threshold and CLR versus mean line utilisation. Such wide variation in results is not present when the the mean line utilisation is graphed against the threshold.

Figure 7.13 graphs the resulting mean line utilisation versus threshold for VP10S1 traffic. These results like those for the TP10S1 traffic (shown in Figure 7.10) give a straight line. The fitted curve is a straight line, the equation is shown in Table 7.6. As compared

	Best fit equation	Squared correlation	Solution for a 10^{-3} CLR
CLR versus Threshold	threshold = 75.7014+ 3.50838 × log(CLR)	0.900	51.47
CLR versus Mean Line Utilisation	mlu = 0.720168+ 0.0316339 × log(CLR)	0.902	0.502
Mean Line Utilisation versus Threshold	mlu = 0.0367509+ 0.00903156× threshold	0.999	-

Table 7.6: Equations of the lines of best fit for the VP10S1 traffic type. Solutions for a CLR of 10^{-3} are listed where applicable.

	Best fit equation	Squared correlation	Solution for a 10^{-3} CLR
Acceptance Boundary in terms of CLR	boundary = 7.57014+ 0.350838 × log(CLR)	0.900	5.147
Mean Line Utilisation versus acceptance boundary	mlu = 0.0367509+ 0.0903156× boundary	0.999	-

Table 7.7: Acceptance boundary relationships derived from Table 7.6. A solution for a CLR of 10^{-3} is listed for the CLR versus acceptance boundary relationship.

with the much smaller values for the squared correlation for the relationships involving CLR, the relationship between mean line utilisation and the threshold value is 0.999. This value was 0.998 for TP10S1 traffic and such high correlations show the tight relationship that exists between mean line utilisation and the selection of the threshold value.

As was done in Section 7.2.3, the relationship between mean line utilisation and the threshold value and the relationship between CLR and the threshold value can be expressed using the acceptance boundary instead of the threshold value. By doing this, the expressions become independent of the PCR of the traffic type. This simple relationship makes easy the task of converting the fitted lines of Table 7.6 into the relationships of Table 7.7. Table 7.7 does not give a relationship between CLR and mean line utilisation as this is unaffected by the conversion from threshold value to acceptance boundary.

The fitted lines of Table 7.7 and Table 7.5 are comparable; these fitted lines represent the acceptance boundary relationships for two different traffic types that have identical ATM Forum TM 4.0 descriptors. The PCR, SCR and MBS of the two traffic types are the same, yet due to differences in the structure of the two sources the resulting acceptance boundaries are different.

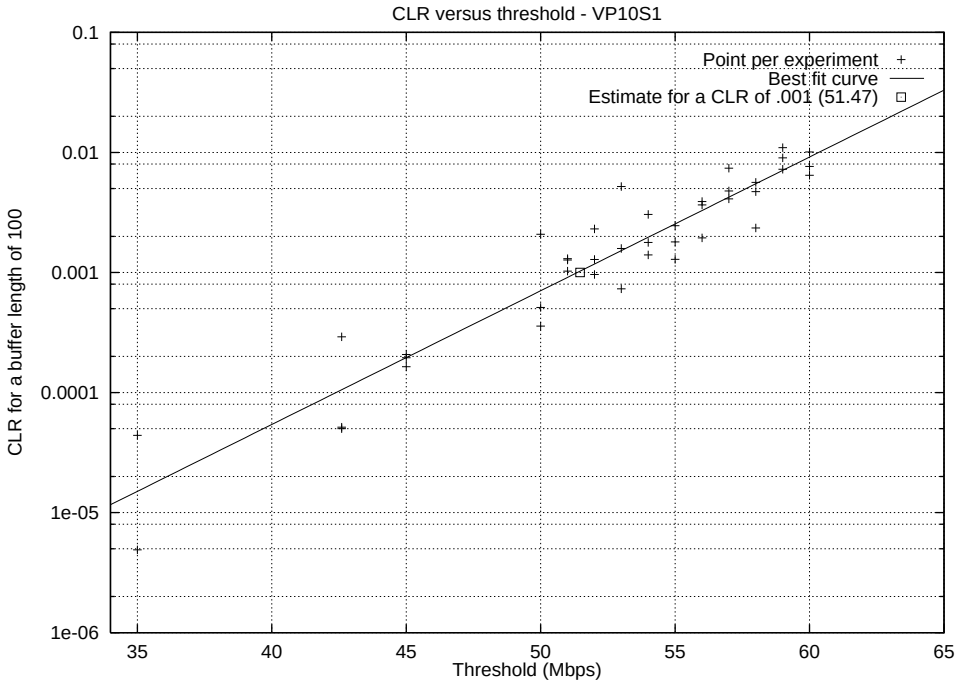


Figure 7.11: CLR versus threshold for VP10S1 experiments.

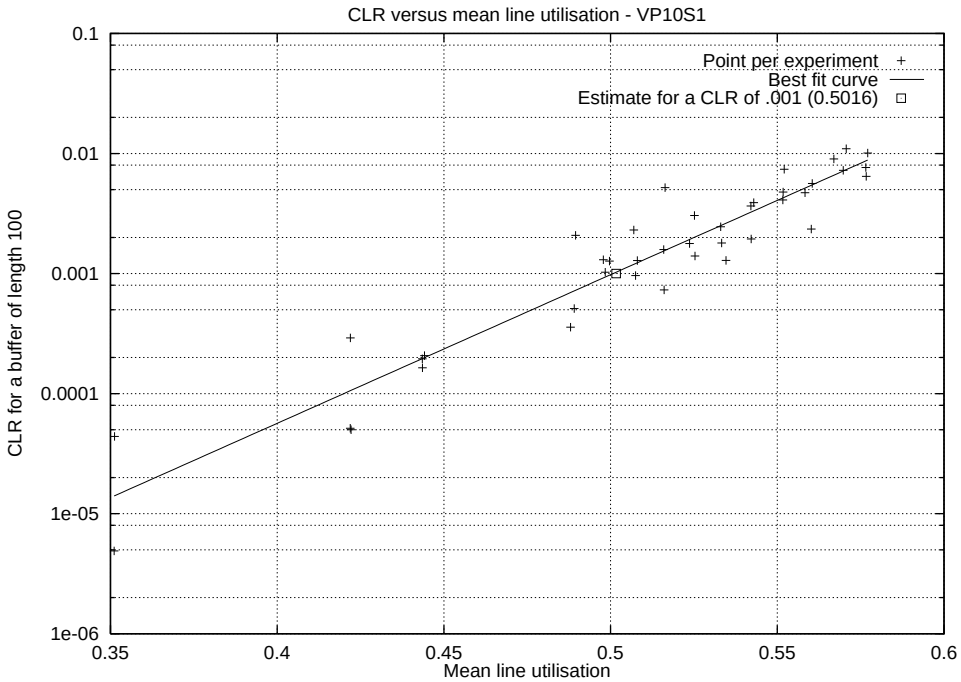


Figure 7.12: CLR versus mean line utilisation for VP10S1 experiments.

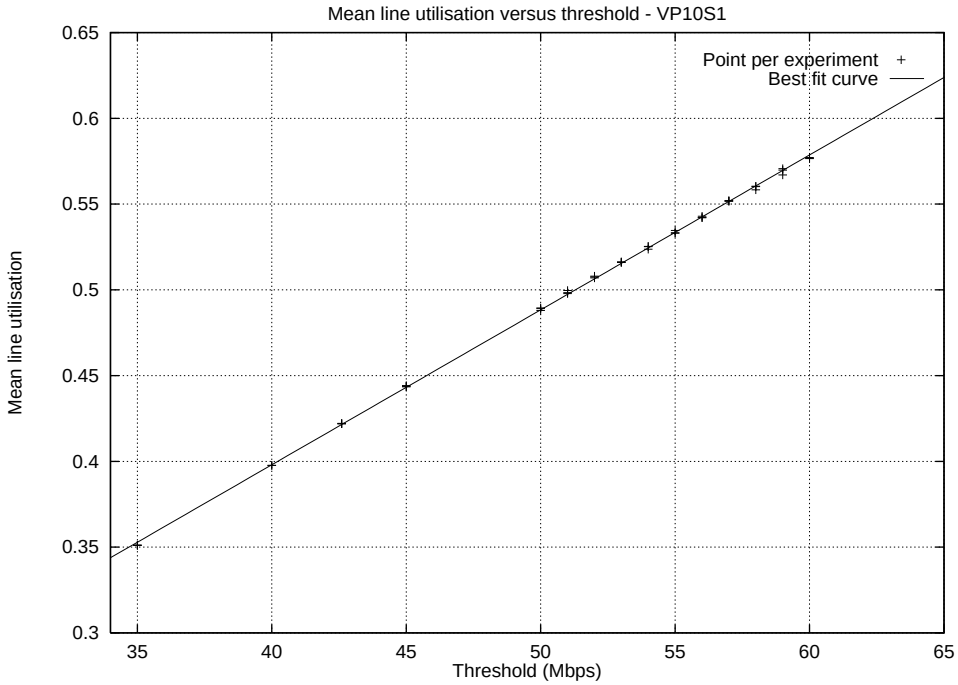


Figure 7.13: Mean line utilisation versus threshold value for VP10S1 experiments.

7.4 Experiments with video traffic type VP5S2

Following the experiments of Section 7.3 where a video stream was used to create a traffic stream with a PCR of 10Mbps, SCR of 1Mbps and MBS of 25 cells; this section uses a traffic type with a different set of parameters: a PCR of 5Mbps, SCR of 2Mbps and MBS of 50 cells per burst. Using a different traffic type enables the threshold-based CAC algorithm to be appraised for an unknown traffic type that is significantly different to traffic types' TP10S1 and VP10S1. The conversion of a stream of video frames into the traffic source VP5S2 is covered in detail in Section 2.3.2. Using the method of the previous section, Section 7.3, the relationships between mean line utilisation, CLR and the threshold value used are established empirically by fitting estimation curves to the results of experiments.

We agreed with our collaborators at BTL that the MCHT for connections carrying traffic type VP5S2 was reduced by half; by doing this each connection transmitted the same mean number of cells into the ATM switch as those connections carrying VP10S1 or TP10S1 but with an MCHT twice as long. The overall parameters for experiments with the VP5S2 traffic type were an MCAR of 10 connections per second, MCHT of 5 seconds per connection. This combined with a PCR of 5Mbps and an SCR of 2Mbps (giving an activity of 0.4) means that the offered load from these connections was 20. Such an increased value of offered load is expected to make significant differences to the relationships this section sets out to establish as compared with Section 7.3.

Figure 7.14 graphs the results for CLR versus the threshold value. This figure also graphs a curve fitted to these results; the equation of this curve is given in Table 7.8. The spread of points of Figure 7.14 have given a moderate fit with a squared correlation of 0.890. Like the results of Section 7.3 such variation in the results themselves could be resulting from the traffic being carried. The VP5S2 traffic type is based on a video stream giving traffic with a periodic structure. This has meant there could be correlation between multiplexed streams of this traffic. It needs to be emphasised this situation will exist in any real-world system carrying multiple video traffic streams encoded in the fashion described in Section 2.3. Using the fitted curve we are able to make a prediction of the threshold that could be used to give a CLR of 10^{-3} for a buffer of 100 cells. The predicted threshold value is 60.57 Mbps. By applying the same empirical technique to the mean line utilisation and CLR results we can obtain a prediction of the mean line utilisation for a CLR of 10^{-3} for a buffer of 100 cells.

The mean line utilisation and CLR results are graphed together in Figure 7.15. This graph also shows a curve fitted to the mean line utilisation versus CLR results; the equation for this curve is given in Table 7.8. In a manner similar to the relationship of CLR and threshold (Figure 7.14), and similar to the results for a different video based traffic source (Figure 7.12), the CLR values vary but still support a relationship between the mean line utilisation and CLR. The variation in CLR results may be due to the periodic nature of the traffic sources. The squared correlation for the fit of the curve to the CLR versus mean line utilisation results is 0.874. Using this curve we can obtain an estimation of the mean line utilisation for a CLR of 10^{-3} for a buffer of 100 cells. This estimation is a mean line utilisation of 0.578.

	Best fit equation	Squared correlation	Solution for a 10^{-3} CLR
CLR versus Threshold	threshold = 73.5662+ 1.88118 × log(CLR)	0.890	60.57
CLR versus Mean Line Utilisation	mlu = 0.685574+ 0.0156647 × log(CLR)	0.874	0.577
Mean Line Utilisation versus Threshold	mlu = 0.0555631+ 0.00861865× threshold	0.999	-

Table 7.8: Equations of the lines of best fit for the VP5S2 traffic type. Solutions for a CLR of 10^{-3} are listed where applicable.

The final relationship for the VP5S2 traffic type is mean line utilisation versus threshold; the results are graphed in Figure 7.16. This figure also shows a curve fitted to the results; this curves' equation is in Table 7.8. In comparison to the high 'spread' of results of the previous two figures, the results of this figure have a tight linear relationship; this curve has a squared correlation of 0.999.

The relationship between mean line utilisation and the threshold value and the relationship between CLR and the threshold value can be expressed using the acceptance boundary instead of the threshold value (as was done in Section 7.2.3 and Section 7.3). By expressing the relationship in terms of the acceptance boundary, the relations become independent of the PCR of the traffic type. The simple relationship between the threshold value and the acceptance value means it is a trivial conversion of the fitted lines of Table 7.8 into the relationships of Table 7.9. Table 7.9 does not give a relationship between CLR and mean line utilisation as it is unaffected by the conversion from threshold value to acceptance boundary. Comparing with the relationships of Table 7.7 which was created using an offered load of 10, Table 7.9 was created for traffic conditions that created an offered load of 20. The result is a three-way relationship between acceptance boundary, offered load and CLR – for two values the third is able to be estimated. In the other case – the mean line utilisation, the acceptance boundary and the offered load share a similar relationship. Building up information about the surface and improving the estimation potential of this three way relationship is the task of the next Section. Section 7.5 uses the same empirical methods as those here to establish the CLR, mean line utilisation and acceptance boundary relationship for a wide range of offered loads.

	Best fit equation	Squared correlation	Solution for a 10^{-3} CLR
Acceptance Boundary in terms of CLR	boundary = 3.6783+ 0.09406 × log(CLR)	0.890	3.029
Mean Line Utilisation versus acceptance boundary	mlu = 0.0367509+ 0.172373× boundary	0.999	-

Table 7.9: Acceptance boundary relationships derived from Table 7.8. A solution for a CLR of 10^{-3} is listed for the CLR versus acceptance boundary relationship.

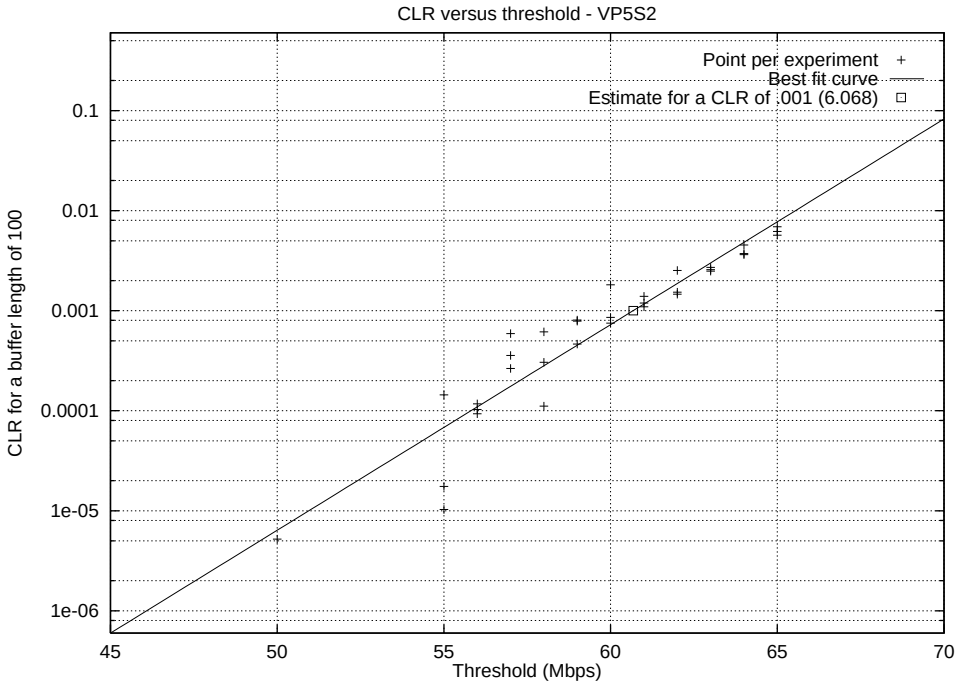


Figure 7.14: CLR versus threshold for VP5S2 experiments.

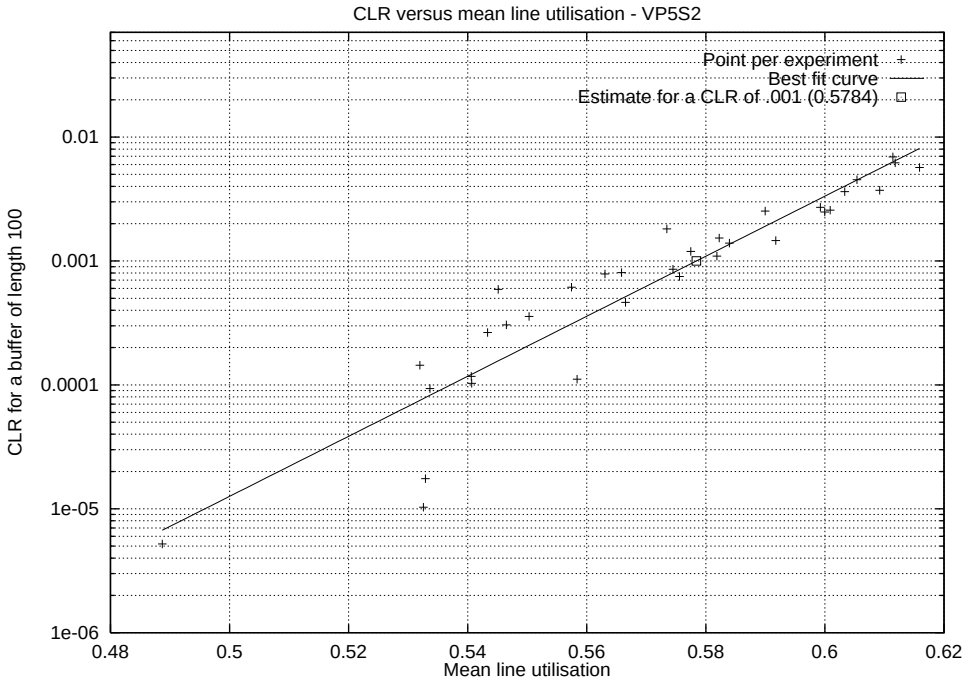


Figure 7.15: CLR versus mean line utilisation for VP5S2 experiments.

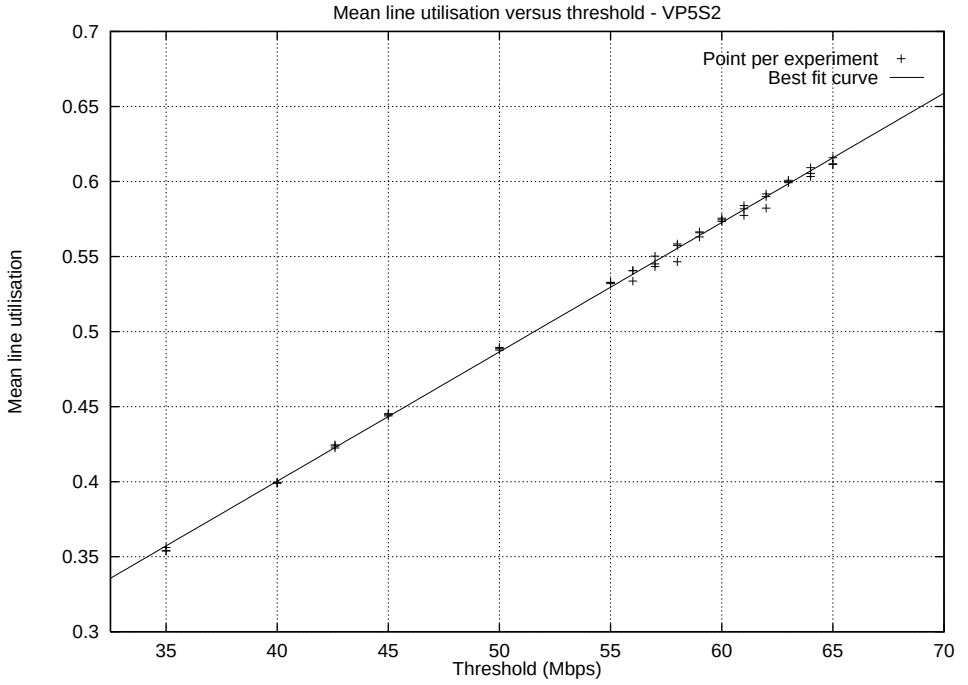


Figure 7.16: Mean line utilisation versus threshold value for VP5S2 experiments.

7.5 Experiments with a mix of traffic types

The objective of this section is to establish the relationship between CLR, mean line utilisation and threshold value (and hence acceptance threshold) when the CAC algorithm is placed under a range of offered load conditions. A range of offered loads were created using a mix of input traffic types. The traffic mix consisted of connections carrying a mix of VP5S2 and VP10S1 traffic types. The process used to create traffic sources VP5S2 and VP10S1 is described in detail in Section 2.3. Table 7.10 gives the nine mixes of inputs into the experiments of this section.

Using the techniques of multiple experiments and curve fitting that were used in previously (such as Section 7.4), experiments to establish the relationship between CLR, mean line utilisation and threshold value, (and thus acceptance boundary) were conducted for each of the offered loads of Table 7.10. These results, combined with the results of Section 7.4 for an offered load of 20 and Section 7.3 for an offered load of 10, give a range of results sufficient to enable the acceptance boundary surface to be predicted empirically.

As discussed in Section 3.4, the acceptance boundary is related to the threshold via a the line capacity, such a relationship means the acceptance boundary can be given in terms that are independent of both the line rate of the network that will use the threshold-based CAC algorithm and the PCR of the connections to be admitted into the system. The line capacity is the number of connections running at PCR that could be accepted into a buffer-less network system of a particular line rate. The line capacity, or more simply the capacity, is based on the line rate, 100Mbps for these experiments, divided by the PCR of the incoming connections. For an experiments where connections are carrying VP10S1, the capacity is 100Mbps $\div$ 10Mbps or 10; If the connections were carrying VP5S2, with a PCR of 5Mbps, the capacity is 100Mbps $\div$ 5Mbps or 20. This situation grows more complex when the incoming connections are carrying a mix of traffic. The capacity is still calculated by dividing the line rate by the PCR, however for a mix of traffic the PCR value used is the mean PCR of the mix of traffic. If the mix of traffic means that half the connections have a PCR of 10Mbps and half with a PCR of 5Mbps, the mean PCR is 7.5Mbps and the resulting capacity is 100Mbps $\div$ 7.5Mbps or approximately 13.333. Table 7.11 gives the capacity for each input traffic mix; the values of capacity are used to convert threshold into values of the acceptance boundary.

The relationship between CLR and the acceptance boundary is shown in Figure 7.17. This figure shows the data from all experiments along with the curves fitted to each set of results. Improving the clarity, Figure 7.18 does not show each individual experimental result, showing the fitted curves only; the formula of the fitted curves are given in Table 7.12. Figure 7.18 also includes the curve resulting from the experiments of Section 7.3 given in Table 7.7 and of Section 7.4 given in Table 7.7. Each curve represents results for a particular set of experiments conducted using a particular offered load. The offered load values are varied by using a number of different mixes of input traffic; the mixes used to obtain the offered load are given in Table 7.10. The concept of an acceptance boundary surface is best shown in Figure 7.19; this graph shows each fitted curve the same as Figure 7.18 but with an additional axis plotting the offered load value. Figure 7.19 also plots the solution

Traffic Details						Total Offered load
VP10S1			VP5S2			
MCAR	MCHT	Offered Load	MCAR	MCHT	Offered Load	
–	–	–	10	5	18	20
1	10	1	9	5	18	19
2	10	2	8	5	16	18
3	10	3	7	5	14	17
4	10	4	6	5	12	16
5	10	5	5	5	10	15
6	10	6	4	5	8	14
7	10	7	3	5	6	13
8	10	8	2	5	4	12
9	10	9	1	5	2	11
10	10	9	–	–	–	10

Table 7.10: Traffic details in experiments with mixes of different traffic types.

Traffic Details						Mean PCR	Capacity
VP10S1			VP5S2				
MCAR	PCR	$\propto$ PCR	MCAR	PCR	$\propto$ PCR		
–	–	–	10	5	5.0	5.0	20.000
1	10	1	9	5	4.5	5.5	18.18
2	10	2	9	5	4.0	6.0	18.182
3	10	3	9	5	3.5	6.5	16.667
4	10	4	9	5	3.0	7.0	14.286
5	10	5	9	5	2.5	7.5	13.333
6	10	6	9	5	2.0	8.0	12.500
7	10	7	9	5	1.5	8.5	11.765
8	10	8	9	5	1.0	9.0	11.111
9	10	9	9	5	0.5	9.5	10.526
10	10	10	–	–	–	10.0	10.000

Table 7.11: Resulting capacity for mixed traffic inputs.

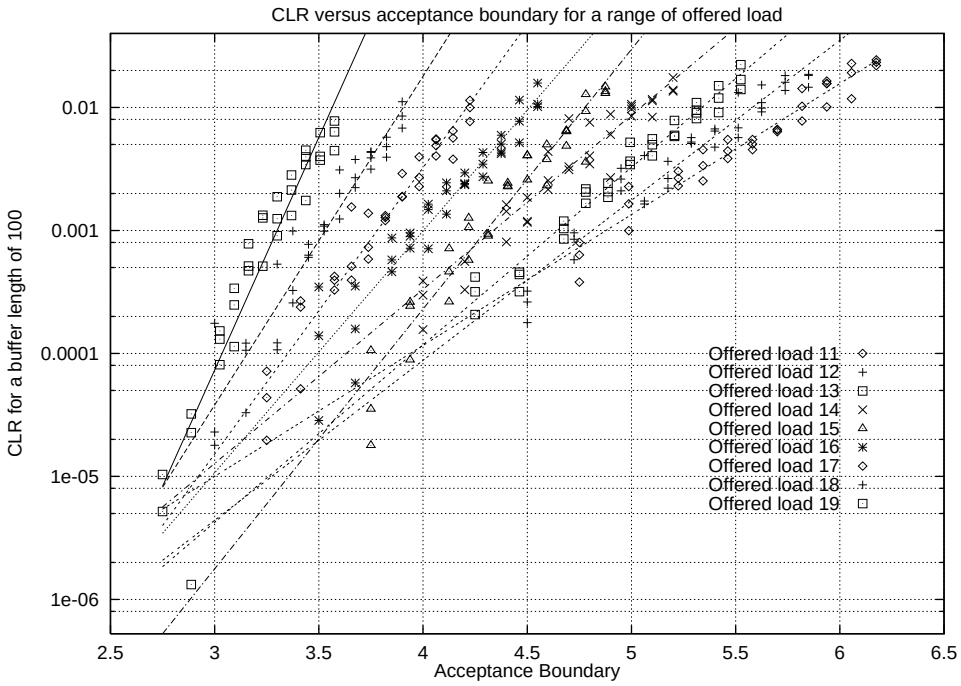


Figure 7.17: CLR versus the Acceptance Boundary for a range of offered loads.

to achieve a CLR of 10^{-3}. The solution for a CLR of 10^{-3} is indicated as the intersection of the acceptance boundary and the plane for a CLR of 10^{-3}. The Figure 7.19 surface shows clearly that for constant values of CLR, acceptance boundary values will vary considerably for any particular offered load. From this graph it becomes clear that values of acceptance boundary that for an offered load of 20 would achieve a given target CLR, would, for an offered load of 10, achieve a target CLR several orders of magnitude less. Such a variable relationship emphasises the importance of establishing correctly the offered load and thus the acceptance boundary.

Figure 7.20 graphs the resulting mean line utilisation and CLR values for experiments conducted with a range of offered load values. Curves fitted to each set of offered load results are also shown on this figure. Figure 7.21 re-graphs only the fitted curves of Figure 7.20, adding the fitted curves for offered loads of 20 and 10 gained from Section 7.3 (Table 7.6) and Section 7.4 (Table 7.8). Each curve plotted represents a set of results resulting for a particular offered load value. Table 7.13 lists the equations of the fitted curves shown in Figure 7.21. The values of squared correlation for the equations are given as an indication of the accuracy of the fit. To better represent the results of Figure 7.21, in Figure 7.22 we add an additional axis to arrive at a CLR surface, values of CLR for combinations of mean line utilisation and offered load. Figure 7.22 also shows the solutions for a CLR of 10^{-3}. The solution for a CLR of 10^{-3} is indicated as the intersection of the CLR surface and the plane for a CLR of 10^{-3}. The surface of Figure 7.22 shows that there is a clear 'flattening' in the relationship between mean line utilisation and CLR as the offered load tends towards 0. This situation can be justified on the basis that lower

Offered load	Line of best fit to results	Squared correlation
10	Acceptance Boundary $= 7.57014 + 0.350838 \times \log(\text{CLR})$	0.8998
11	Acceptance Boundary $= 7.58779 + 0.387266 \times \log(\text{CLR})$	0.9496
12	Acceptance Boundary $= 6.98844 + 0.310514 \times \log(\text{CLR})$	0.9301
13	Acceptance Boundary $= 6.67066 + 0.291902 \times \log(\text{CLR})$	0.9701
14	Acceptance Boundary $= 6.25271 + 0.271453 \times \log(\text{CLR})$	0.8851
15	Acceptance Boundary $= 5.6205 + 0.189808 \times \log(\text{CLR})$	0.9231
16	Acceptance Boundary $= 5.39094 + 0.199729 \times \log(\text{CLR})$	0.9043
17	Acceptance Boundary $= 4.98668 + 0.174363 \times \log(\text{CLR})$	0.9347
18	Acceptance Boundary $= 4.55666 + 0.148708 \times \log(\text{CLR})$	0.9152
19	Acceptance Boundary $= 4.00169 + 0.102611 \times \log(\text{CLR})$	0.8938
20	Acceptance Boundary $= 3.67831 + 0.0940589 \times \log(\text{CLR})$	0.8904

Table 7.12: Fit lines for the acceptance boundary in terms of CLR.

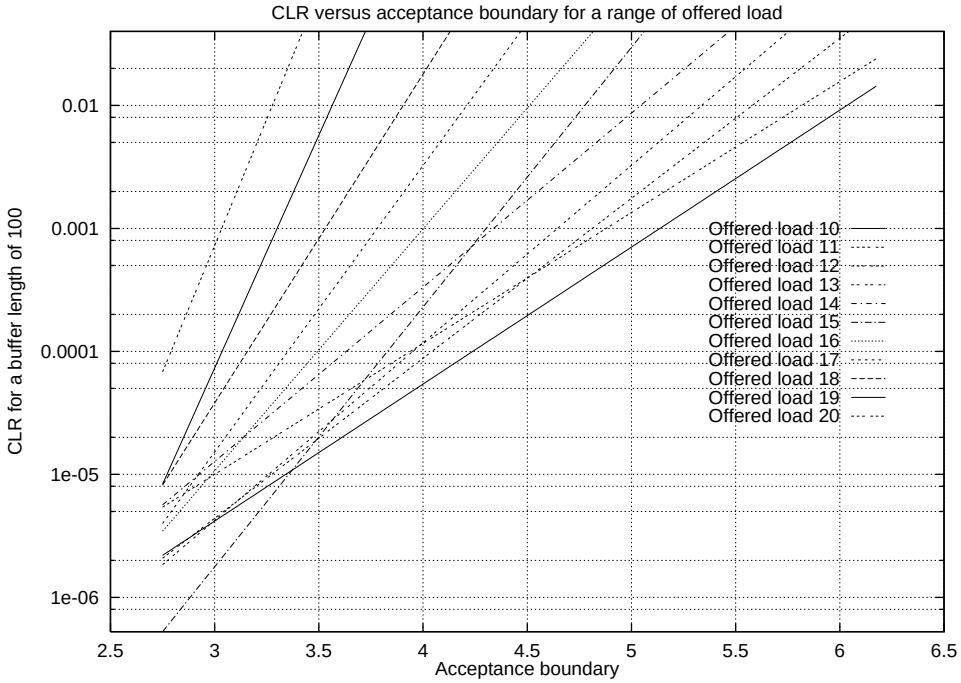


Figure 7.18: CLR versus the Acceptance Boundary for a range of offered loads – fitted curves only.

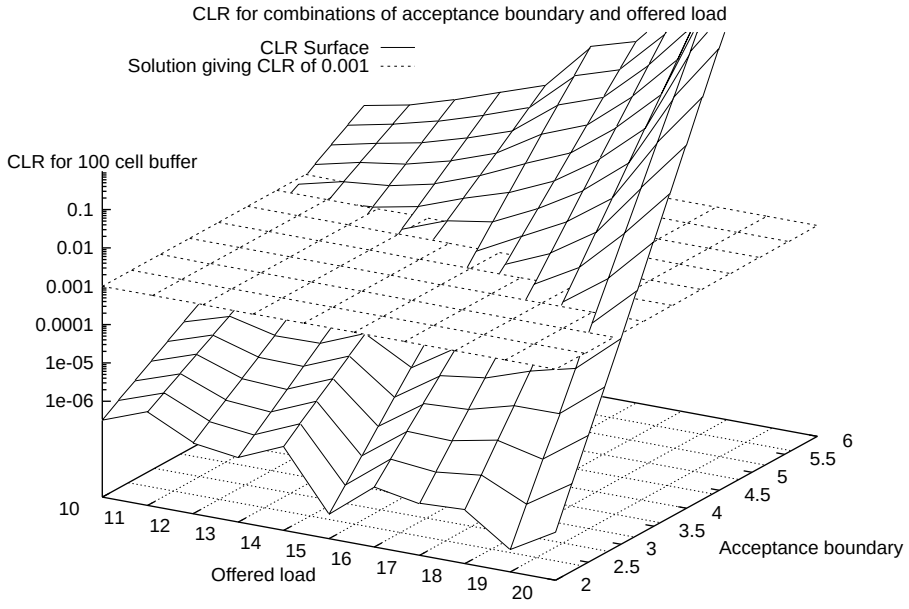


Figure 7.19: CLR for combinations of the offered load and the acceptance boundary.

Offered load	Line of best fit to results	Squared correlation
10	$\text{mlu} = 0.720168 + 0.0316339 \times \log(\text{CLR})$	0.9017
11	$\text{mlu} = 0.747527 + 0.0353043 \times \log(\text{CLR})$	0.9501
12	$\text{mlu} = 0.726647 + 0.0295939 \times \log(\text{CLR})$	0.9305
13	$\text{mlu} = 0.729429 + 0.0290807 \times \log(\text{CLR})$	0.9725
14	$\text{mlu} = 0.724944 + 0.0284254 \times \log(\text{CLR})$	0.8805
15	$\text{mlu} = 0.698873 + 0.0213798 \times \log(\text{CLR})$	0.9277
16	$\text{mlu} = 0.715421 + 0.0239083 \times \log(\text{CLR})$	0.8852
17	$\text{mlu} = 0.716366 + 0.0228287 \times \log(\text{CLR})$	0.9268
18	$\text{mlu} = 0.707386 + 0.0208573 \times \log(\text{CLR})$	0.9071
19	$\text{mlu} = 0.66841 + 0.0137352 \times \log(\text{CLR})$	0.7415
20	$\text{mlu} = 0.685574 + 0.0156647 \times \log(\text{CLR})$	0.8739

Table 7.13: Fit lines for the mean line utilisation in terms of CLR.

offered load values will result in less active traffic, that is traffic that is less likely to cause significant CLR as a result of abrupt changes in the traffic characteristics.

The final relationship, that of acceptance boundary and mean line utilisation, is revealed in results graphed in Figure 7.23. The fitted curves for experiments with each value of offered load are also shown in this figure. The equations of the fitted curves are given in Table 7.14. Alongside the fitted curve equations, the squared correlation is also given as an indication of the accuracy of the fitted curve. Figure 7.21 graphs just the fitted curves graphed in Figure 7.23 – also shown are the fitted curves for the acceptance boundary versus mean line utilisation relationship for offered loads of 10 and 20. The fitted curves for offered loads of 10 and 20 were developed in Section 7.3 (Table 7.7) and Section 7.4 (Table 7.9) respectively. To more clearly represent the results of Figure 7.24, in Figure 7.25 an offered load axis is added giving a surface of mean line utilisation for the acceptance boundary and offered load values. Figure 7.25 gives evaluated contours across the surface to make the surface shape clearer. While the relationship appears to be near linear, the surface plot of Figure 7.25 makes clearer the surface curve and shape.

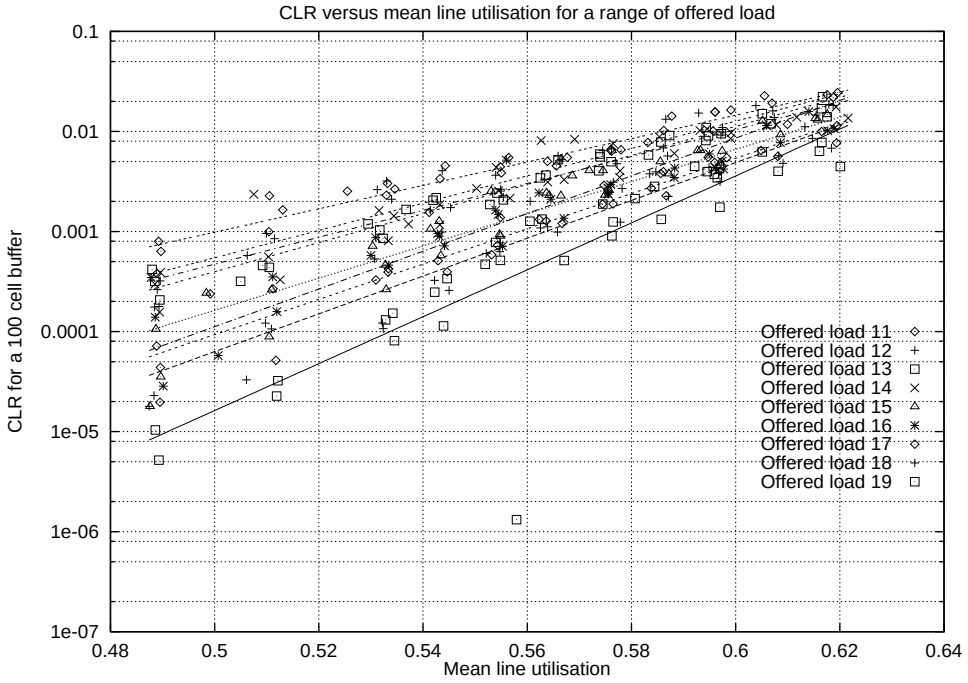


Figure 7.20: CLR versus the mean line utilisation for a range of offered loads.

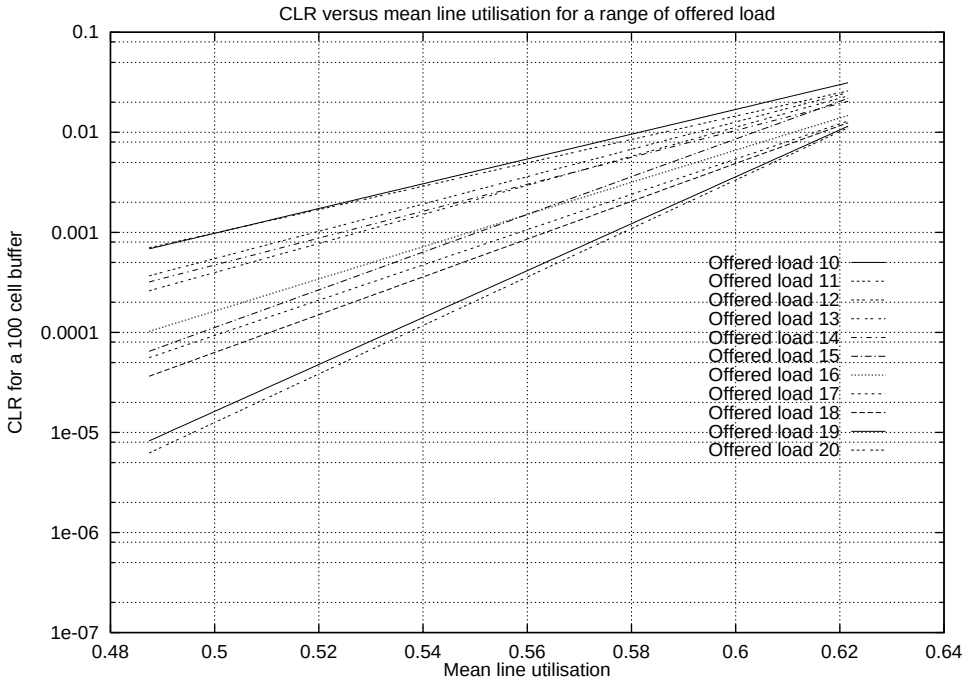


Figure 7.21: CLR versus the mean line utilisation for a range of offered loads – fitted curves only.

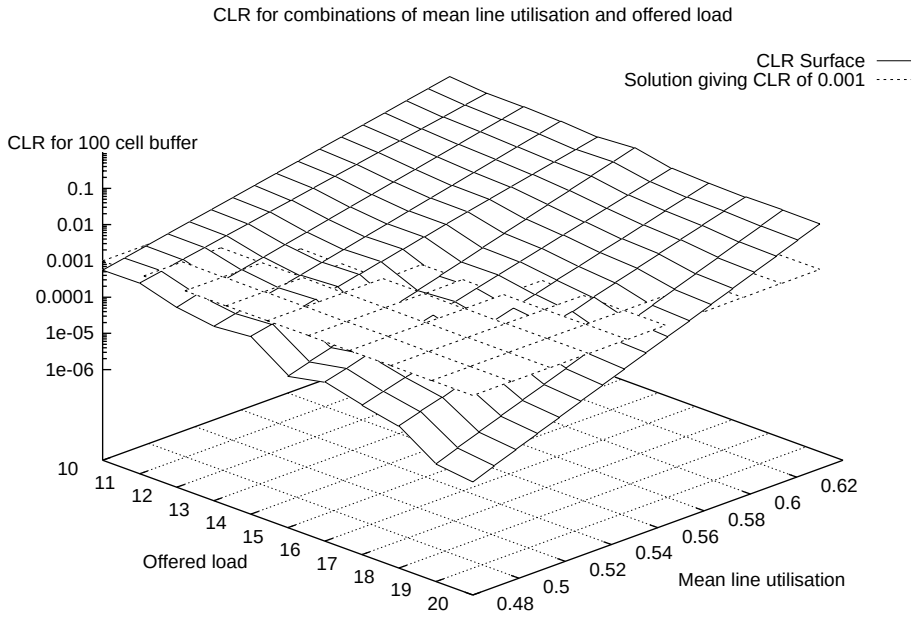


Figure 7.22: CLR for combinations of the offered load and the mean line utilisation.

Offered load	Line of best fit to results	Squared correlation
10	mlu = 0.0367509+ acceptance boundary ×0.0903156	0.9997
11	mlu = 0.057847+ acceptance boundary ×0.0907952	0.9925
12	mlu = 0.0611546+ acceptance boundary ×0.0952021	0.9982
13	mlu = 0.0681077+ acceptance boundary ×0.0989731	0.9894
14	mlu = 0.0826334+ acceptance boundary ×0.102058	0.9449
15	mlu = 0.0752791+ acceptance boundary ×0.110476	0.9668
16	mlu = 0.0786717+ acceptance boundary ×0.117613	0.9449
17	mlu = 0.0644696+ acceptance boundary ×0.130666	0.9876
18	mlu = 0.0699767+ acceptance boundary ×0.139775	0.9843
19	mlu = 0.0983388+ acceptance boundary ×0.144448	0.9519
20	mlu = 0.0555631+ acceptance boundary ×0.172373	0.9986

Table 7.14: Fit lines for the mean line utilisation in terms of the acceptance boundary.

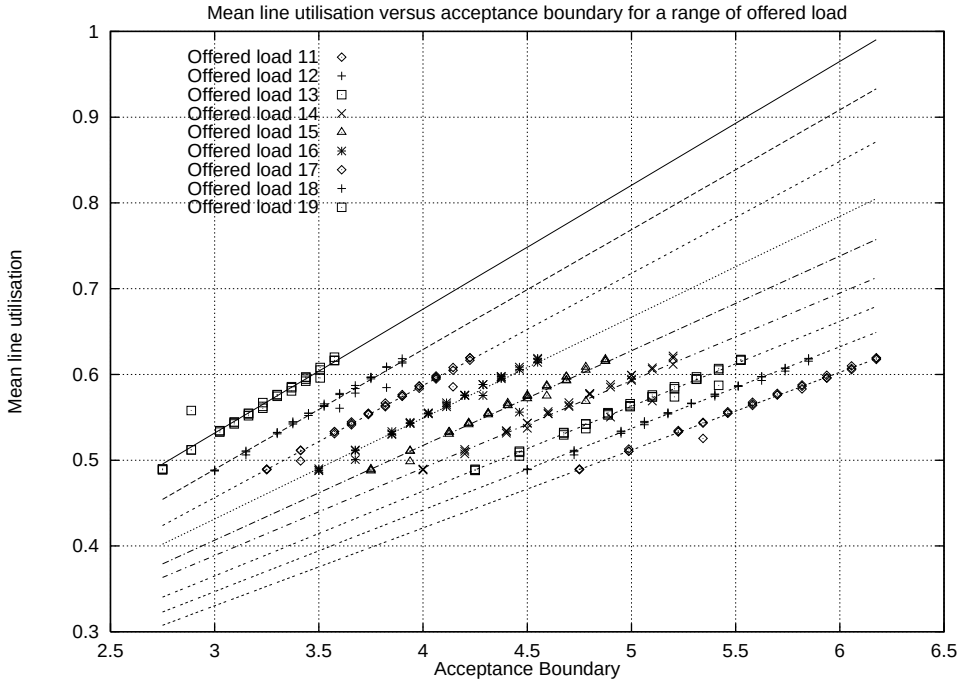


Figure 7.23: Mean line utilisation versus acceptance boundary for a range of offered loads.

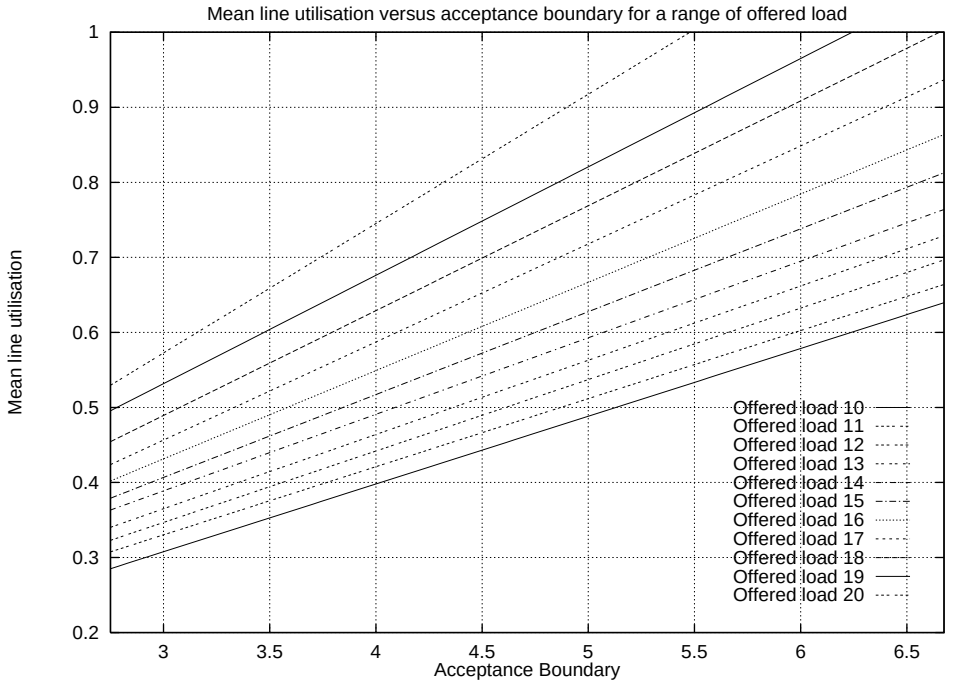


Figure 7.24: CLR versus the mean line utilisation versus acceptance boundary for a range of offered loads – fitted curves only.

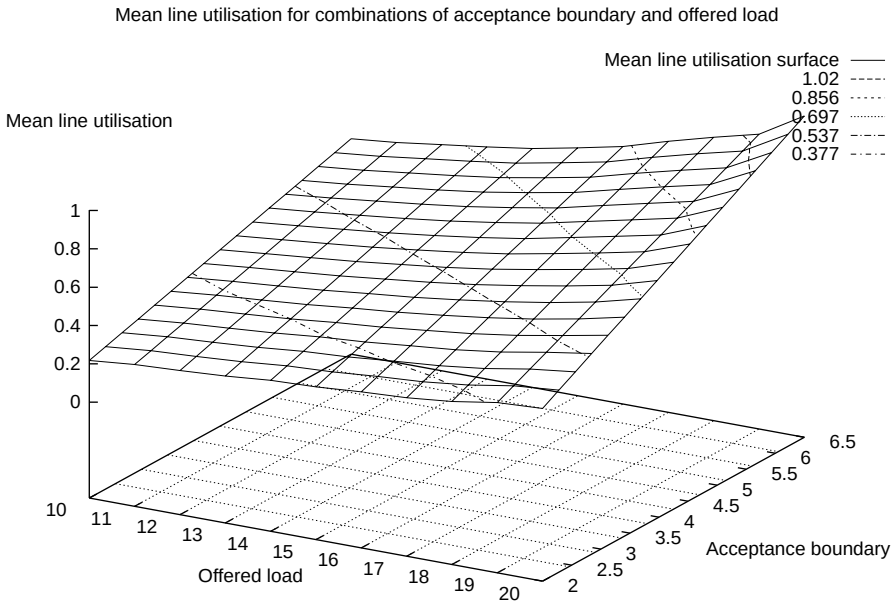


Figure 7.25: Mean line utilisation for combinations of the offered load and the acceptance boundary.

7.6 Comparison with other CAC algorithms

The CAC evaluation environment was designed to allow comparison of experiments using the same CAC algorithm under different conditions of connection load and traffic type. Additionally, the CAC evaluation environment was designed to allow different CAC algorithms to be compared under similar conditions of connection and traffic. This section gives a comparison of results made using the BT adaptive algorithm against a ATM Forum mandated CAC algorithm and a comparison against a CAC algorithm under development at Cambridge. The theory of the BT adaptive algorithm or Key algorithm is described in Section 3 and implementation specific details are described in Section 7.1. In comparison the ATM Forum CAC algorithm is much simpler. The most significant difference between algorithms in this comparison is that the mean results of experiments made using the BT adaptive CAC algorithm are compared with results gained from single runs of experiments using alternative CAC algorithms. Multiple runs of the alternative CAC algorithms were, while desirable, unable to be performed in the available time.

The Peak Rate Allocation CAC algorithm, mandated by the ATM Forum as the most basic CAC algorithm, makes no use of information about the current line utilisation. The Peak Rate Allocation CAC algorithm is highly pessimistic, using the declared PCR of each new connection to calculate a worst-case utilisation; this worst-case allocation is compared with the available line capacity and a decision for admission is made on the basis of there being enough capacity remaining to allow the new connection admission into the network. If there is not enough remaining capacity in the network, the new connection attempt is rejected. In contrast the Measure CAC algorithm uses some parameter information from new connections combined with information obtained by measuring the network utilisation.

The Measure CAC algorithm is a simplified implementation of the large-deviation [15, 16] mechanism discussed in [13, 12, 14, 4]. The implementation uses a regularly-calculated value for the effective bandwidth required by the connections currently in the network in order to achieve a given CLR for a given buffer size. This is combined with the declared PCR of new connections to decide if the network can support the new connection request. A connection request will be admitted into the system if its PCR will fit into the bandwidth remaining along with the PCR values of other new connections that are yet to be taken into account by the regular recalculation of the effective bandwidth estimate. The bandwidth remaining is the total line capacity minus the effective bandwidth requirement. In this way the measure algorithm attempts to maintain a particular CLR for the line and adapts as the connection load and/or traffic types change.

The Measure algorithm is naïve in that the aggregate traffic utilisation over the lifespan of the network is used in calculations of effective bandwidth. A "correct" implementation of Measure would use the traffic utilisation over the lifespan of each connection. The result of this difference is that the Measure algorithm records a history of traffic fluctuations over the course of the network links' lifespan. This record includes short-term network transients, such as those experienced during the start-up period, and will, thus, affect the long-term accuracy of the estimation. One direct consequence of this is that the Measure algorithm can behave like a peak-rate allocation algorithm if the effective bandwidth re-

quirement remains at a high level for a long-enough period. The algorithm can, under certain circumstances, be confused by large transients in the input traffic which are maintained in the history and the effective bandwidth estimate can remain high even after the connections that caused the transients have long been pulled-down. Notwithstanding this, the Measure algorithm forms a direct comparison with the Key algorithm as an alternative approach that also uses measurements of the line.

The Key CAC algorithm has limitations on the range of values of the measurement period, as discussed in Section 3. In this naïve implementation of the Measure CAC algorithm, the selection of a parameter over which measurements are summed, the block length, is subject to the same limitations on its range of values. The precise behaviour of the Measure algorithm is in some ways dependent on the selection of this blocking parameter. Where available, two different sets of results for two different values of the block length are given.

7.6.1 Theoretical traffic source TP10S1

The experiments of Section 7.2 were duplicated for each of the alternative CAC algorithms. These experiments involved new connections arriving with an MCAR of 10 connections per second and an MCHT of 10 seconds per connection; the connections are carrying traffic type TP10S1, this theoretical traffic source detailed in Section 2.2.1 has an SCR of 1 Mbps, a PCR of 10Mbps and an MBS of 25 cells. Table 7.15 presents the results obtained for a number of different CAC algorithms; also this table lists a number of estimates: estimate results derived from results in this report and estimate results derived from theoretical models.

Using the acceptance boundary of 4.26, originally specified by our colleagues at BTL to achieve a CLR of 10^{-3}, we obtain the results listed first in Table 7.15. As predicted by the experiments of Section 7.2, the Key algorithm is slightly optimistic in its admission statistics giving a CLR that exceeds the target of 10^{-3}. Following the result for the Key CAC algorithm is the outcome of an experiment made using the same traffic and the same connection load but with the Peak Rate Allocation CAC algorithm.

The results in Table 7.15 for the Peak Rate Allocation algorithm reflect the pessimistic approach taken by this algorithm. While the CLR was 0, no cells being lost, the line utilisation is only 0.08 and the call acceptance/rejection ratio is 0.1. Two sets of results for the Measure CAC algorithm are reported next.

The two Measure CAC algorithm results arise from the different behaviour the Measure algorithm will give if it uses different values for the block length parameter. It can be seen that the two different values of block length give similar mean line utilisation, connection acceptance/rejection ratios and figures for mean connection in progress. However the two different block lengths have resulted in two significantly different CLR values. The results for this naïve implementation of Measure have returned significantly better figures for connection acceptance/rejection ratios and mean line utilisation figures yet still managed CLR figures that indicate the algorithm was overly pessimistic in admission and wasted some available network resources by achieving a CLR less than the desired target of 10^{-3}.

The Measure algorithm performed even better than the results predicted by Section 7.2 using a refined acceptance boundary value. Using the relationships developed empirically in Section 7.2.3 and given in Table 7.4 a value for the acceptance boundary needed to achieve a CLR of 10^{-3} can be calculated. The mean line utilisation can then be predicted for this particular acceptance boundary. The acceptance boundary value of 4.12 gives a predicted mean line utilisation of 0.409.

Theoretical models can also give estimates for the mean connections in progress a figure roughly comparable with the connection accept/reject ratio if the experiment has a sufficiently high MCAR. For experiments conducted using the CAC evaluation algorithm, the figure of the mean connections in progress can be calculated for the course of an experiments. In the table an estimation for mean connections in progress is given from several theoretical estimators: Guérin [21], Elwalid [17] and Buffet & Duffield [7]. It is interesting to note the significant divergence in the results for commonly used theoretical models.

7.6.2 Varied values of offered load

In Section 7.5 the relationships between CLR, mean line utilisation and the acceptance boundary (as used in the Key CAC algorithm) were investigated for a range of values of offered load. A comparison between the performance of the Key CAC algorithm and other CAC algorithms for a similar range of offered loads is presented here. The range of values of offered load are created by mixing connections carrying traffic types VP10S1 and VP5S2 in varying ratios; the relationship between the ratios of incoming connections and the offered load presented to the CAC algorithm is detailed in Table 7.10. A theoretical comparison is not available for connections carrying the video traffic sources VP10S1 and VP5S2, hence no theoretically derived results are presented here.

In addition to results gained using the Peak Rate Allocation mechanism, the comparison of Table 7.16 gives two sets of results (using different block lengths) for the naïve implementation of the Measure CAC algorithm. The values for the Key algorithm are estimations calculated using the relationships developed in Section 7.5. Several sets of results for the Measure algorithm show poor performance, this is due to the naïvity of the Measure implementation as discussed at the beginning of Section 7.6. An example of the problems created by the naïve implementation of the Measure CAC algorithm can be seen with the results for an offered load of 12. Using an offered load of 12, the Measure algorithm with a block length of 100ms has a very poor connection acceptance ratio, this seems to be due to the Measure algorithm disallowing all new connections due to a early line measurements causing an overly high long-term measurement of the bandwidth requirement of the current connections.

It is important to note that time did not allow more than one run of each of the Measure CAC algorithm based experiments and only one of the Peak Rate Allocation CAC algorithm based experiment. In comparison the results for the Key CAC algorithm are estimations and because of this, show the 'smoothing' that has resulted from the numerous experiments run in Section 7.5 to develop the relationships used to calculate the values

Algorithm name	CLR for a 100 cell buffer	Connection accept ratio	Mean connections in progress	Mean line utilisation
Key (acceptance = 4.26)	1.38×10^{-3}	0.535	53.0	0.420
Peak Rate Allocation	0	0.100	9.89	0.080
Measure (block length=10ms)	6.82×10^{-4}	0.554	53.8	0.522
Measure (block length=100ms)	8.83×10^{-4}	0.568	54.9	0.549
Experimentally derived estimate				
Key (acceptance = 4.12)	1.00×10^{-3}	0.526	51.96	0.409
Theoretical model estimations				
Guérin	1.00×10^{-3}	–	36.92	–
Elwalid	1.00×10^{-3}	–	42.98	–
Buffett & Dufield	1.00×10^{-3}	–	45.20	–

Table 7.15: Comparative set of results for experiments with similar traffic conditions.

used here.

Table 7.16: Comparison of CAC algorithms for a variety of offered loads.

Offered Load	CAC used	CLR for a 100 cell buffer	Mean line utilisation	Total connection accept ratio	VP10S1 connection accept ratio	VP5S2 connection accept ratio
AB is Acceptance Boundary for Key estimate; BL is Block Length for Measure.						
10	Key AB=5.15	1.00×10^{-3}	0.50	0.58	–	–
	Peak Rate	0	0.08	0.20	–	–
	Measure BL=10ms	1.17×10^{-3}	0.30	0.21	–	–
	Measure BL=100ms	3.90×10^{-4}	0.28	0.25	–	–
11	Key AB=4.91	1.00×10^{-3}	0.50	0.63	0.63	0.63
	Peak Rate	0	0.11	0.22	0.19	0.46
	Measure BL=10ms	1.21×10^{-5}	0.18	0.17	0.01	0.19
	Measure BL=100ms	8.44×10^{-6}	0.48	0.44	0.01	0.49
12	Key AB=4.84	1.00×10^{-3}	0.52	0.64	0.64	0.64
	Peak Rate	0	0.16	0.24	0.19	0.45
	Measure BL=10ms	2.98×10^{-5}	0.23	0.22	0.01	0.27
	Measure BL=100ms	3.08×10^{-4}	0.33	0.01	0.01	0.01
13	Key AB=4.65	1.00×10^{-3}	0.53	0.66	0.66	0.66
	Peak Rate	0	0.16	0.26	0.19	0.43
	Measure BL=10ms	8.58×10^{-6}	0.23	0.21	0.13	0.24
	Measure BL=100ms	6.94×10^{-5}	0.55	0.51	0.13	0.68
14	Key AB=4.38	1.00×10^{-3}	0.53	0.65	0.65	0.65
	Peak Rate	0	0.19	0.28	0.18	0.44
	Measure BL=10ms	3.88×10^{-4}	0.16	0.15	0.01	0.24
	Measure BL=100ms	6.51×10^{-5}	0.37	0.33	0.08	0.50
15	Key AB=4.31	1.00×10^{-3}	0.55	0.67	0.67	0.67
	Peak Rate	0	0.22	0.31	0.18	0.44
	Measure BL=10ms	9.48×10^{-6}	0.15	0.13	0.01	0.26
	Measure BL=100ms	2.35×10^{-5}	0.33	0.30	0.01	0.59
16	Key AB=4.01	1.00×10^{-3}	0.55	0.66	0.66	0.66
	Peak Rate	0	0.24	0.32	0.17	0.42
	Measure BL=10ms	2.11×10^{-6}	0.21	0.20	0.14	0.31
	Measure BL=100ms	3.81×10^{-5}	0.54	0.52	0.45	0.62
17	Key AB=3.78	1.00×10^{-3}	0.56	0.66	0.66	0.66
	Peak Rate	0	0.27	0.34	0.17	0.42
	Measure BL=10ms	6.79×10^{-5}	0.19	0.18	0.15	0.26
	Measure BL=100ms	4.11×10^{-5}	0.48	0.45	0.41	0.56
18	Key AB=3.53	1.00×10^{-3}	0.56	0.67	0.67	0.67
	Peak Rate	0	0.32	0.36	0.17	0.41
	Measure BL=10ms	4.61×10^{-5}	0.18	0.17	0.16	.24
	Measure BL=100ms	5.66×10^{-5}	0.32	0.31	0.30	0.37

continued...

Offered Load	CAC used	CLR for a 100 cell buffer	Mean line utilisation	Total connection accept ratio	VP10S1 connection accept ratio	VP5S2 connection accept ratio
19	Key AB=3.29	1.00×10^{-3}	0.57	0.67	0.67	0.67
	Peak Rate	0	0.34	0.38	0.16	0.41
	Measure BL=10ms	1.75×10^{-4}	0.23	0.21	0.16	0.65
	Measure BL=100ms	2.96×10^{-5}	0.46	0.44	0.43	0.48
20	Key AB=3.03	1.00×10^{-3}	0.58	0.75	–	–
	Peak Rate	0	0.36	0.40	–	–
	Measure BL=10ms	1.32×10^{-5}	0.16	0.16	–	–
	Measure BL=100ms	9.49×10^{-5}	0.08	0.18	–	–

To aid comparison, the results from each CAC algorithm experiment and from the Key estimation are compared graphically in Figures 7.26(a),7.26(b), 7.27. The results of CLR versus offered load results are graphed in Figure 7.26(a), this graph shows that the Key algorithm was using a CLR of 10^{-3} as an input, hence a straight line when the CLR is compared with the offered load. The Measure results for CLR show no distinct relationship between CLR and offered load, while it is difficult to make any judgement from a single set of results, it is expected that the CLR would be maintained regardless of the connection load. The Measure algorithm using a 10ms blocking factor achieved a considerably more stable, albeit lower CLR value than the experiment with a block length of 100ms. The Peak Rate Allocation CAC algorithm returned a CAC of 0 for each offered load, as a result the figures for Peak Rate Allocation are not plotted in this figure.

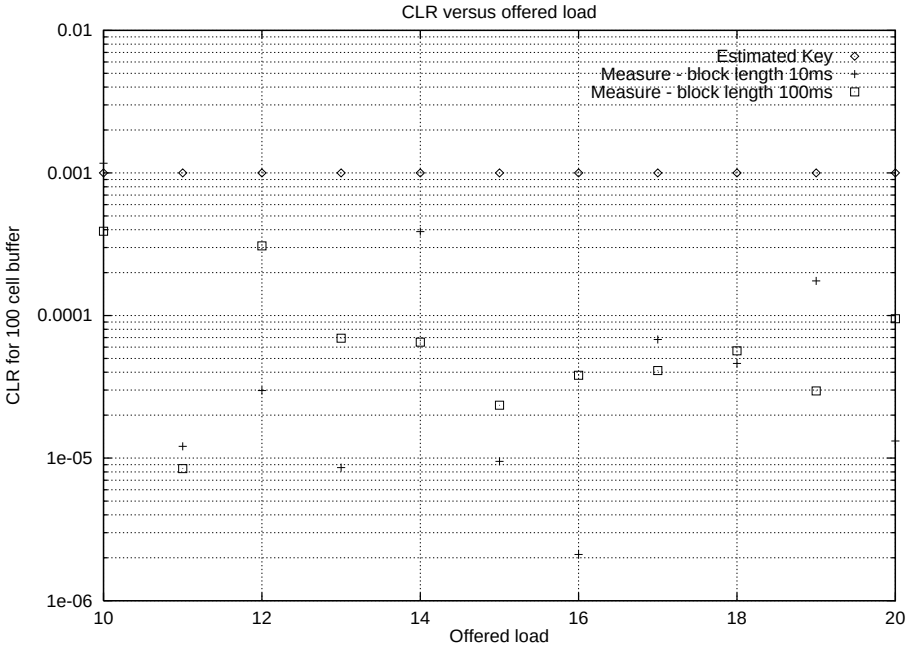
The Peak Rate Allocation CAC behaviour is dictated solely the relationship between the declared PCR of incoming connections and the overall capacity on the line. In these experiments the line capacity was 100Mbps and for an offered load of 10 all connections carry traffic with a PCR of 10Mbps therefore ten of these connections will be admitted as a maximum as a direct result the mean line utilisation will be 10 times the SCR of each of the admitted connections, thus the mean line utilisation for an offered load of 10 is a figure near to 10Mbps. For the situation where offered load is 20, connections carry traffic with a PCR of 5Mbps thus 20 connections can be admitted. 20 connections will mean 20 times the SCR of this traffic, (2Mbps) this results in a mean line utilisation approaching 40Mbps. Such a situation is shown clearly in Figure 7.26(b). This figure also shows a relationship that is close to linear between the mean line utilisation and the offered load for the Peak Rate Allocation CAC algorithm.

The results for the Key algorithm also share a relationship that is close to linear between the mean line utilisation and the offered load. This situation is expected because previous sections (7.2.3 7.3 7.4) showed a linear relationship could be established between the mean line utilisation and the threshold, and thus the acceptance boundary. Referring to Table 7.10, a range of acceptance boundaries (and thus a range of thresholds) were used to obtain the desired CLR. It is anticipated that Figure 7.26(b) shows a linear relationship

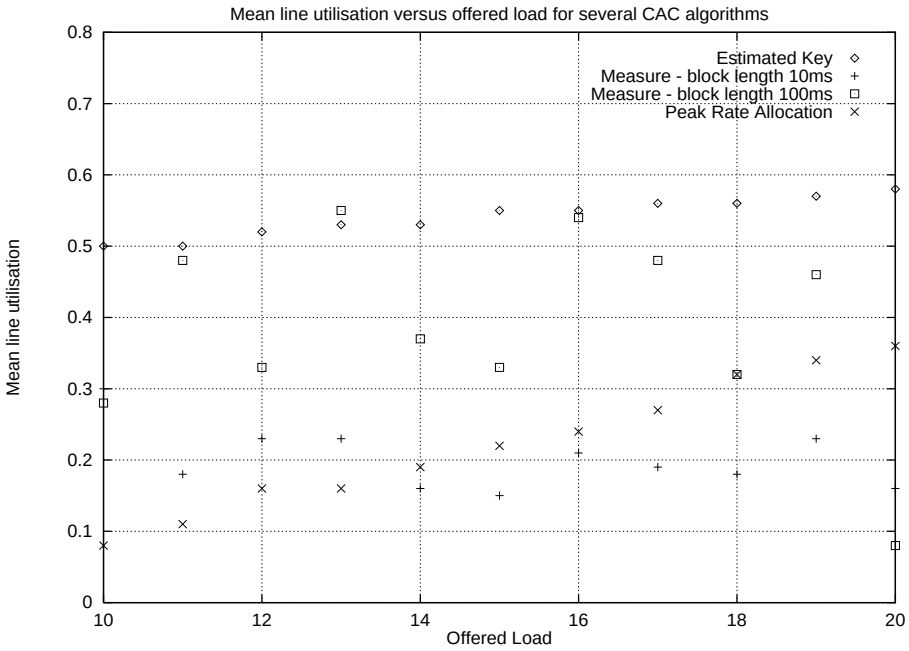
between the offered load (and hence the threshold) and the mean line utilisation for the Key CAC algorithm. In comparison the sample Measure results in Figure 7.26(b) show no distinct relationship between mean line utilisation and offered load. Such a situation is not surprising 'cause the Measure algorithm is attempting to obtain a particular CLR value, the mean line utilisation achieved is only a side-effect. The only note on the Measure results is the different mean line utilisation figures for Measure CAC results due to different values of the block length; these results indicate the importance of the selection of this parameter.

Figure 7.27 contains a graphical representation of the connection accept/reject ratios for all connections (Figure 7.27(c)), just the connections carrying VP10S1 traffic (Figure 7.27(b)) and the connections carrying just the VP5S2 traffic (Figure 7.27(a)). The ratios for connection specific traffic requests are ratios of successful connections of that type to the total connection attempts for that type. It is expected that the Key CAC algorithm will show no bias in which connections it admits; a lack of bias should be represented by the same response to offered load for each of the three graphs of Figure 7.27. The figures show an identical response for each of the three ratios (total connection attempts and connection attempts by type). This situation results from the Key algorithm not using any information from the declared parameters; the only input into the Key algorithm is the instantaneous measurement of the line load. It is anticipated the Peak Rate Allocation CAC algorithm is biased towards new connections with the lowest PCR value. The results in Figure 7.27 show that a bias exists in all values of offered load where there is a combination of connections (values of offered load from 11 to 19) bias exists for most values of the offered load. Comparing the results graphed in Figure 7.27(b) and those graphed in Figure 7.27(a) it can be seen that the connections carrying a load of 5Mbps (VP5S2) have twice the chance of being admitted as compared with connections carrying a load of 10Mbps (VP10S1). However, while the bias in admissions appears to be explained by this behaviour it must be remembered that the mix of connections for various values of offered load (Table 7.10) has meant the connections carrying traffic with a higher PCR have an MCHT of half the period of the MCHT for the traffic with a lower PCR. With a lower MCHT, it is conceivable that more admissions of connections carrying VP5S2 traffic can be accepted because there is a higher probability a 5Mbps "block" of space being available in the link. The actual behaviour of the Peak Rate Allocation CAC algorithm could be determined by using traffic of two different PCR values being carried by connections with the same arrival characteristics, MCAR and MCHT however this was well outside the original scope of the project and its investigation was left to future work.

In contrast to the Peak Rate Allocation and Key CAC algorithms, the Measure CAC algorithm produces the most erratic output. The Measure algorithm exhibits behaviour that is similar to the Peak Rate Allocation mechanism and also similar to the Key algorithm. Once the effective bandwidth estimation is a significant proportion of the total line bandwidth available, the Measure algorithm employs a Peak Rate Allocation style of admission for the remaining bandwidth resource. One immediate effect of this is that the Measure algorithm using a block length significantly smaller than the MCAR will give admissions that are not biased towards new connections with small PCR values. If however the Measure effective bandwidth estimation is large enough, a bias away from new

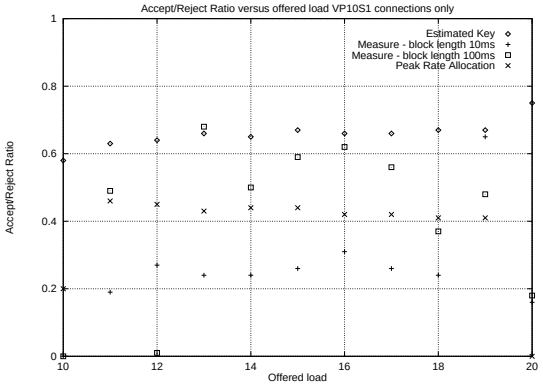


(a) CLR for a range of offered load.

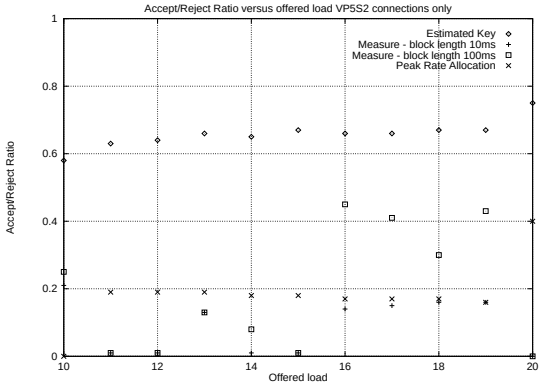


(b) Mean line utilisation for a range of offered load.

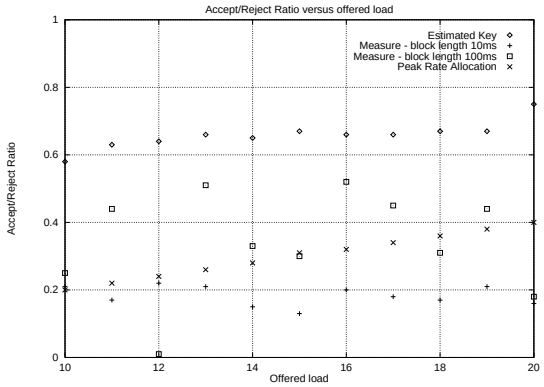
Figure 7.26: Results for a range of offered load.



(a) VP10S1 connections only



(b) VP5S2 connections only



(c) All connections

Figure 7.27: Connection Accept/Reject ratios for a range of offered load.

connections with a particular PCR will be introduced. This situation can become critical, if the effective bandwidth estimation becomes large enough no new connections will be admitted by the CAC algorithm regardless of their PCR values. This characteristic is shown in Figure 7.27 for an offered load of 12. During this experiment the effective bandwidth estimation became so large that it caused all new connections to be refused regardless of their PCR. Results for the Measure CAC algorithm form an interesting comparison but it must be emphasised the Measure CAC algorithm is a naïve implementation using aggregate line bandwidth rather than per connection bandwidth as was originally intended by its creators. One direct impact of using per-connection bandwidth measurements would be that long term historical anomalies affecting the effective bandwidth calculations would be eliminated and the overall stability of the effective bandwidth calculations and the CAC algorithm would be improved as a result.

8 Conclusion

Interest in CAC algorithms stems from the need for a network user and a network provider to establish an agreement on the QoS for a connection the user wishes to have admitted into the network. This study has presented results of experiments culminating in the evaluation of the BT adaptive CAC algorithm and the comparison of that CAC algorithm with several other CAC algorithms.

Section 2 discussed the two sources, one theoretical and one derived from video stream data, used in this CAC evaluation. Also discussed were the issues involved in the encoding of video for transport over a network.

Section 3 followed by detailing the functioning of CAC algorithms based upon thresholding techniques as well as how these algorithms adapt to the varying of network conditions. This discussion lead to explicit coverage of aspects of the BT adaptive CAC algorithm.

An initial set of experiments to evaluate parameters of the CAC mechanism as well as parameters of the evaluation experiments were conducted using the environment described in Section 4. This section also reports on unique aspects of the experimental test environment and the procedures used to conduct the experiments.

Section 5 documented results created using the test environment of Section 4. These results were principally to enable our BTL collaborators to check effective bandwidth calculations and to investigate the relationship between CLR and buffer overflow. Additionally the initial experiments allowed the level of accuracy required for each experiment to be established; this established directly the minimum number of cells to give a balance between the run time of experiments and obtaining statistically significant values for CLR. To achieve an experiment with a satisfactory running time but with a statistically significant number of cells, experiments with 1×10^8 cells transmitted through the run was consider to be satisfactory. Finally the initial experiments measured the relationship between traffic source independence and the complementary queue length distribution (Complementary CDF). This set of measurements established the need for a new style of traffic generator. Finally Section 5 justified no further experiments using the theoretical traffic source TP20S5 described in Section 2.2.2.

Section 6 described the experimental configuration used in the evaluation of the BT adaptive threshold based CAC algorithm. In addition to a description of the CAC test-rig, and a discussion of performance aspects of the test-rig, Section 6 evaluates the performance, repeatability and adaptability of the completed test-rig.

Section 7 concluded this report with a coverage of the results gained using the BT adaptive CAC algorithm. The thresholding values created by our BTL collaborators for a traffic system with theoretical traffic sources were shown to be pessimistic when put into practical application. Section 7 established empirically the relationship between CLR, thresholding value and the mean line utilisation. From these relationships it was possible to predict a threshold value that would achieve the desired CLR values. Section 7 then continued reporting on results to empirically establish the relationship between CLR, thresholding value and the mean line utilisation for two video-derived traffic sources as a comparison with the relationships estimated for the theoretical traffic source. This section

reports on results obtained using a range of values of offered load, obtained by using a blend of the two video-derived traffic sources; this enabled empirical establishment of the affect on the relationships between CLR, thresholding value and the mean line utilisation that result from varied values of offered load. Section 7 finished with a series of comparisons between the BT adaptive CAC algorithm and two other CAC algorithms under similar conditions. While this comparison was less conclusive in its results, it was able to show important characteristics of the BT adaptive CAC algorithm, particularly as compared with the characteristics of other CAC algorithms.

Future Work

The scope for future work is very broad, and there are a number of areas of potential development.

One would be to extend the Key CAC mechanism to a more-adaptive system that can take into account the mix of traffic types. There are a number of algorithms discussed in [19] that would enable the threshold mechanism to have the more-adaptive capability that it was originally intended to possess. Such a direction would require the measurement rig be able to determine the line utilisation resulting from a particular type of traffic. That is a logical extension to the current environment and would give a worthy insight into the fully adaptive algorithm. A modification to the Key CAC algorithm suggested by Bean [3] would be another logical direction to take. The extension Bean suggests is to use a more rigid connection rejection policy giving the Key CAC algorithm some historical information upon which it can base decisions.

It is a current intention to use the CAC test-environment to evaluate and compare a range of other CAC algorithms from the literature (as mentioned in Section 6.3.3) as well as comprehensively test the performance of the Measure algorithm [13, 12, 14, 4]. Finally Section 7.6 mentions further work in evaluating the behaviour of better established CAC algorithms such as Peak Rate Allocation would make an interesting study.

Thanks

The work of this report has been possible only with the much-appreciated input of several colleagues at both the British Telecom, Martlesham Heath Laboratories and the University of Cambridge, Computer Laboratory. Much gratitude is extended to Gary Walley for his guidance and input on this project throughout its execution. Many thanks to Peter Key who has provided an invaluable source of information on the algorithm and the environment in which it should be tested.

A debt is owed to developers of the Nemesis operating system, and Austin Donnelly in particular, for making possible the construction of robust and highly adaptable traffic generators. Thank you to Robin Fairbairns for looking over drafts of this report and his typesetting assistance.

Finally, thanks to Ian Leslie, Tim Granger and Neil Stratford for their valuable input over the course of this project.

9 Glossary

CAC Connection Admission Control

CBR Constant Bit Rate

CDF Queue length distribution

CLR Cell Loss Ratio

Complementary CDF Complementary queue length distribution

ICT Inter-Cell Time

Key CAC algorithm BT Adaptive CAC algorithm detailed in [19, 23, 30].

Load/Offered Load a method of normalising combinations of MCAR, MCHT, PCR and SCR of incoming connections detailed in Section 3.2.

MBS Mean Burst Size

MCAR Mean Connection Arrival Rate, the rate of arriving connection attempts.

MCHT Mean Connection Holding Time

nrt-VBR non-real-time Variable Bit Rate

PCR Peak Cell Rate

QoS Quality of Service

rt-VBR real-time Variable Bit Rate

RPC Remote Procedure Call

SCR Sustained Cell Rate

Working Set the size of the ICT list used in a physical traffic controller - limited by the physical size of memory of a traffic generator.

References

[1] ATM Forum. *Traffic Management Specification, Version 4.* ATM Forum/95-0013R8, October 1995.

[2] Abdella Battou. Connections Establishment Latency: Measured Results. *ATM-Forum T1A1.3/96-071*, October 1996.

[3] N. G. Bean. Robust connection acceptance control for ATM networks with incomplete source information. *Annals of Operations Research*, 1994.

[4] Nils Bjorkman, Alexander Latour, Aziz Miah, Simon Crosby, Ian Leslie, John Lewis, Raymond Russell, and Fergal Toomey. Exploring the queueing behaviour of ATM switches. In *Proceedings Performance '96*, EPFL, Lausanne, Switzerland, October 1996. IFIP WG 7.3.

[5] Richard Black and Simon Crosby. Experience and results from the implementation of an ATM socket family. In *Proceedings USENIX Winter Technical Conference*, San Francisco, January 1994.

[6] Richard Black, Ian Leslie, and Derek McAuley. Experiences of building an ATM switch for the Local Area. In *Proceedings ACM SIGCOMM*, volume 24(4), September 1994.

[7] E. Buffet and N.G. Duffield. Exponential upper bounds via martingales for multiplexers with markovian arrivals. *J. Appl. Prob.*, 31:1049–1061, 1992.

[8] CCITT SG XVIII. Recommendation I.362 – BISDN ATM adaptation layer functional description. Technical report, TD22, Geneva, January 1990.

[9] CCITT SG XVIII. Recommendation I.363 – BISDN ATM adaptation layer specification. Technical report, TD22, Geneva, January 1990.

[10] Simon Crosby. The causes and magnitude of CDV in ATM networks: A practical study using Fairisle. Technical report, University of Cambridge Computer Laboratory, 1995. Deliverable 1, BT research project.

[11] Simon Crosby. Performance management in ATM networks. Technical Report 393, Cambridge University Computer Laboratory, May 1995. Ph.D. dissertation.

[12] Simon Crosby, Meriel Huggard, Ian Leslie, John Lewis, Fergal Toomey, and Cormac Walsh. Bypassing modelling: Further investigations of entropy as a traffic descriptor in the Fairisle ATM network. In *Proceedings WATM'95 First Workshop on ATM Traffic Management*, Ecole Nationale Superieure des Telecommunications, Paris, December 1995. IFIP W.G. 6.2.

[13] Simon Crosby, Ian Leslie, John Lewis, Neil O'Connell, Raymond Russell, and Fergal Toomey. Bypassing modelling: an investigation of entropy as a traffic descriptor in the Fairisle ATM network. In *Proceedings Twelfth UK Teletraffic Symposium*, Windsor, England, March 1995. IEE, Springer Verlag.

[14] Simon Crosby, Ian Leslie, John Lewis, Raymond Russell, Meriel Huggard, and Brian McGurk. Predicting effective bandwidths of ATM and ethernet traffic. In *Proceedings Thirteenth UK Teletraffic Symposium*, Strathclyde University, Glasgow, March 1996. IEE.

[15] N. G. Duffield, J. T. Lewis, Neil O'Connell, Raymond Russell, and Fergal Toomey. The entropy of an arrivals process: a tool for estimating QoS parameters of ATM traffic. In *Proceedings of the 11th UK Teletraffic Symposium*, Cambridge, March 1994.

[16] N.G. Duffield, J.T. Lewis, N. O'Connell, R. Russell, and F. Toomey. Entropy of ATM traffic streams. *IEEE Journal on Selected Areas in Communications, Special issue on advances in the fundamentals of networking – part 1*, 13(6), August 1995.

[17] A.I. Elwalid, D. Mitra, and T.E. Stern. Statistical multiplexing of Markov modulated sources: Theory and computational algorithms. In *Proceedings of the 13th International Teletraffic Congress*, Copenhagen, June 1991.

[18] M. W. Garrett and W. Willinger. Analysis, modeling and generation of self-similar vbr video traffic. In *Proceedings ACM SIGCOMM 94*, pages 269–280, London, UK, August 1994.

[19] R.J. Gibbens, F.P. Kelly, and P.B. Key. A decision theoretic approach to call admission control in ATM networks. *IEEE Journal on Selected Areas in Communications, Special issue on advances in the fundamentals of networking – part 1*, 13(6), August 1995.

[20] Matthias Grossglauser and David Tse. A framework for robust measurement-based admission control. *ACM SIGCOMM Computer Communications Review*, 27(4):237–248, Oct 1997.

[21] R. Guérin, H. Ahmadi, and Naghshineh. Equivalent capacity and its application to bandwidth allocation in high speed networks. *IEEE JSAC*, 9:968–981, 1991.

[22] Joseph Y. Hui. Resource allocation for broadband networks. *IEEE JSAC*, 6(9):1598–1608, December 1988.

[23] Peter Key. Connection admission control in ATM networks. *BT Technology Journal*, 13(3), July 1995.

[24] Didier Le Gall. MPEG: a video compression standard for multimedia applications. *Communications of the ACM*, 34(4):47–58, April 1991.

[25] Ian Leslie, Derek McAuley, Richard Black, Timothy Roscoe, Paul Barham, David Evers, Robin Fairbairns, and Eoin Hyden. The design and implementation of an operating system to support distributed multimedia applications. *IEEE Journal on Selected Areas in Communication*, 1996.

[26] I.M. Leslie and D. McAuley. Fairisle: An ATM Network for the Local Area. In *Proceedings ACM SIGCOMM*, Zürich, September 1991.

[27] Nemesys Research Limited, editor. *AVA Owner's Manual*. Nemesys Research Limited, September 1996.

[28] Douglas Niehaus, Abdella Battou, Andrew McFarland, Basil Decina, Henry Dardy, Vinai Sirkay, and Bill Edwards. Performance Benchmarking of ATM Networks. *IEEE Communications*, 35(8):134–143, August 1997.

[29] Vern Paxson. Fast, approximate synthesis of fractional gaussian noise for generating seld-similar network traffic. *ACM SIGCOMM Computer Communications Review*, 25(5):5–18, Oct 1997.

[30] J. W. Roberts, editor. *Broadband Network Teletraffic (Final report of Action COST242).* Springer-Verlag, August 1996. ISBN 3-540-61815-5, Chapter 5.

[31] Oliver Rose. Statistical properties of mpeg video traffic and their impact on traffic modeling in ATM systems. Technical Report 101, University of Wuerzburg, Institute of Computer Science Research Report Series, February 1995.

[32] Mark Shand. Measuring system performance with reprogrammable hardware. Technical Report PRL-RR-19, DEC Paris Research Laboratory, August 1992.

[33] *ATM Document Collection.* Systems Research Group, University of Cambridge Computer Laboratory, 3rd (The Blue Book) edition, March 1994. SRG Technical Note available as: http://www.cl.cam.ac.uk/Research/SRG/bluebook.html.