Teletraffic Modelling, Analysis and Synthesis of a Generic Broadband Multi-service Access Protocol

Milosh V. Ivanovich *B.E. (Hons), MComp*
Department of Computer Science
Monash University

Thesis submitted for examination
for the degree of
Doctor of Philosophy in Computing

December 1997

Table of Contents

Abstract

Achieving efficient design and dimensioning of broadband access networks, which represent the major cost component of today's global telecommunications network infrastructure, requires deep insight into the operation and peculiarities of access network protocols under realistic traffic conditions. The focus of this thesis is on a proposed generic multi-service access protocol, called Fair Centralised Priority Reservation, or F-CPR. We provide an extensive teletraffic study of F-CPR's performance and behaviour under realistic traffic conditions, including a set of models exploring signalling-based protocol deadlock. As is often the case, such a teletraffic study provides insight into the protocol's peculiarities and leads to the synthesis of significant extensions which are either part of, or in addition to, the medium access control (MAC) protocol.

The first part of the thesis presents a detailed simulation study of the F-CPR MAC protocol, under realistic traffic conditions based on Ethernet LAN traces, with particular attention being paid to intra- and inter-station correlation. F-CPR is found to exhibit maxmin throughput fairness when loaded by traffic of both a Poissonian and self-similar nature; positional fairness is also determined, with no physical location within the HFC access network providing a station with more than its fair share of medium access. Perhaps counterintuitively, highly correlated traffic is found to improve overall F-CPR utilisation somewhat, as a result of the protocol's contention-free "piggybacked" bandwidth reservation feature. This feature is also found to be very beneficial in postponing, or sometimes completely avoiding, signalling-channel congestion collapse, which is shown to be possible when a large number of stations with extreme inter-station correlation and light individual station load, transmits very small messages. It is also demonstrated that excluding disaster scenarios, and extreme inter-station correlation, the F-CPR performs very close to its Ideal Multiplexer benchmark, and hence can be modelled as an Ideal Multiplexer for a wide range of traffic parameters, with increasing accuracy as the number of stations becomes higher. Another important discovery in our simulation study is that the overall distribution of the generated traffic load interacts with the F-CPR's stop and wait nature between messages, to significantly impact global utilisation performance. Thus if only a small fraction of the active stations generates a large proportion of the traffic load, overall utilisation is significantly lower than if the load is more evenly distributed among the stations.

The second component of this thesis is the development of a suite of disaster scenario (deadlock) models and their analysis by means of a discrete-time Markov chain technique, and the introduction of the concept of comparing between practical and theoretical stability. We provide a detailed set of conditions which are shown to lead to practical instability and deadlock, and which depend on such factors as signalling channel error probability, the profile of signalling traffic, and properties of the contention resolution algorithm being used on the signalling channel. Finally, we propose and test three new signalling channel capacity allocation schemes, with a view to extending the usable region of the F-CPR protocol, by avoiding deadlock under as wide a range of conditions as possible. We identify the best-performing of these three schemes and provide insight into the reasons for its success.

In the third major thrust of our work, we develop a new multi-priority scheduling scheme for the Head-End controller, based on pre-emptive principles. In addition, we propose a mechanism for the generation of multi-priority traffic from an (unprioritised) existing trace, based on so-called priority groups, each of which have member stations that may only transmit messages of a given priority level. We compare and contrast the performance of our new multi-priority scheduling scheme, and prioritised traffic generation mechanism, to an existing scheduling scheme and a random-hash based generation mechanism. In particular, our new scheduling scheme is shown to perform better in minimising average access delay of high priority traffic under realistic intra- and inter-station correlation conditions. In our multi-priority scheduling scheme analysis, an interesting counterintuitive finding is that under extreme inter-station correlation, the average access delay of the high priority messages may significantly decrease with load. This phenomenon was found to be caused by a combined positive effect comprised of: (i) the non-discriminatory nature of the contention-based signalling channel at lower loads, and (ii) the simultaneous preferential treatment of, and increased "piggybacked" bandwidth reservation usage rate by, the high priority traffic with increasing load.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any other university, and to the best of my knowledge contains no material previously published or written by another person except where reference has been made in the text of the thesis.

Signed:

Milosh Vladimir Ivanovich

Department of Computer Science
Monash University

Acknowledgements

I wish to thank my supervisor Dr. Moshe Zukerman for his vision and guidance during my studies, and for always setting aside the time to meet, in what is a truly unforgiving schedule. I particularly acknowledge the methods and philosophy he has imparted to me; these methods have enabled me to modularise complex engineering networks and systems into smaller, tractable components for modelling and analysis. My thanks also go out to Dr. Ron Addie for always providing useful comments, particularly in the domain of self-similar traffic modelling, and for his suggestions for improving the work.

I would also like to thank Chatschik Bisdikian, Dolors Sala and John Limb, for their deep insights about Hybrid Fibre/Coax access networks and IEEE 802.14 standards development. The contribution of Dr. Rasti Slosiar is also acknowledged for his ideas and suggestions about simulation techniques and measurements.

I also greatly thank my father, Dr. Vladimir D. Ivanovic, for proof-reading this thesis, and mother, Slavica Ivanovic for her help regarding presentation style and printing.

I thank my fiancee Gordana for her love and encouragement during my postgraduate years. Finally, and most importantly, my parents deserve an enormous thank-you for their continued and unwavering moral support, and most of all, unconditional love throughout my entire life.

I dedicate my thesis to them: Mami, Tati i Goci.

1. Introduction

An increasing amount of multimedia content on the Internet is fuelling the bandwidth needs of both the core and access components of the network. However, it is the access part of current telecommunications networks, which represents the most expensive and complex component of the global network infrastructure. Therefore, efficient design and dimensioning of the access network leads to significant cost savings and return on investments to telecommunications carriers, and provides better quality of service to customers. Achieving efficient design and dimensioning requires deep insight into the operation and peculiarities of access network protocols under realistic traffic conditions. In this thesis, we focus on a generic multi-service access protocol and provide an extensive teletraffic study of its performance and behaviour under realistic traffic conditions, including a disaster scenario. As is often the case, such a teletraffic study provides insight into the protocol's peculiarities and leads to the synthesis of significant extensions which are either part of, or in addition to, the generic protocol.

In analysing protocol performance, it is very important to predict the statistical characteristics of the traffic loading that the protocol will be subject to. It is thus desirable to use a traffic trace, be it measured or model generated, which is representative of true operating conditions and hence may be considered realistic. Realistic traffic traces often explore the limits and even some peculiar artefacts of the protocol under study, as we shall later see in Chapter 6. As Chapter 2 explains, the lack of agreement in the current state of the art literature about which model accurately and consistently captures the short- and long-term dependence characteristics as well as queueing performance of traffic, prompts us to opt for a measured trace in our protocol testing. Apart from being able to capture the full suite of traffic behaviour and characteristics, a measured traffic trace can be directly related to end user actions or application usage at certain instants, and can thus give more intuition into protocol performance under such conditions.

Our generic medium access control (MAC) protocol, called Fair Centralised Priority Reservation (F-CPR) is a particular implementation of an earlier protocol, CPR, which was proposed by Sala and Limb [SALA 96a]. CPR, and hence F-CPR, is consistent with the majority of features of the dominant MAC proposal currently before the IEEE 802.14 committee ([BISD 96a], [DOSH 96]), which is developing standard specifications for the physical and MAC layer components of a protocol for Hybrid Fibre/Coax (HFC) access networks. HFC is one of today's most well-known network technologies and architectures for the provision of broadband access services, many of which are surveyed in Chapter 3. HFC access is widely considered to be an attractively positioned "bridging technology" between legacy narrowband modems and a purely fibre path to the customer premises. Their attractiveness both in terms of total provisioning cost and the fact that fibre is rolled out closer to customer will make HFC access networks a sensible bridging technology until the Internet backbone network and application servers both reach speeds that are able to cope with the speed of a "fibre access pipe" going all the way to the customer's residence.

The principles of combining contention- and reservation-based bandwidth allocation in most proposed HFC MAC protocols, are also applicable to the MAC protocol for wireless asynchronous transfer mode (WATM) access networks. While there are certain unavoidable medium-related differences, many

operational and architectural characteristics are shared by the two types of MAC protocol. As a result, the F-CPR protocol's fundamental similarity to the WATM MAC protocol allows us to interpret the results of the work presented in this thesis, not only in an HFC access network setting, but also in the case of a wireless ATM system.

Of great interest to the MAC protocol engineer is the protocol's resiliency to conditions potentially leading to a disaster scenario, often termed deadlock. Such an event must be avoided, through both pre-emptive and dynamically adaptive measures, since it usually means that congestion collapse of the signalling, and hence data, channels has taken place and stations' access delay has exceeded practically usable levels. Chapter 4 presents a number of teletraffic models for such a disaster scenario, in the context of the F-CPR MAC protocol.

The importance of meeting widely varying quality of service requirements when it comes to the provision of multimedia services, makes it essential for an HFC MAC protocol to provide efficient and flexible support for multiple traffic priorities, as discussed in Chapter 3. In addition, similarly effective multi-priority scheduling schemes are needed at the Head-End central controller, to operate externally to, but in close conjunction with, an HFC MAC protocol. Two schemes for the provision of multiple priorities in conjunction with F-CPR, along with associated issues, are described in Chapter 5.

Having motivated the work presented in this thesis, we shall now provide a summary of its contributions to the state of the art.

The main contributions of this thesis are:

- By means of a detailed simulation of the F-CPR MAC protocol, under realistic traffic conditions based on campus LAN Ethernet traces, and giving special consideration to intra- and inter-station correlation, discovery of important protocol characteristics and peculiarities. Presented in Chapter 6, the findings include:
    - Consistent with known results for Poisson traffic, the F-CPR, when loaded by highly self-similar traces, is shown to exhibit throughput fairness based on maxmin principles, and total impartiality to a station's physical position within the HFC access network.
    - A highly correlated realistic traffic load, which would intuitively be expected to adversely affect Ideal Multiplexer performance in comparison to a Poisson traffic load, actually somewhat improves overall utilisation in F-CPR. This counterintuitive phenomenon occurs because the highly correlated nature of the real trace causes a greater system-wide probability of at least one non-empty queue, thus enabling F-CPR's contention-free "piggybacked" bandwidth reservation feature, which relies on non-empty queues, to be used more often.
    - Excluding disaster scenarios, and extreme inter-station correlation, the F-CPR performs very close to its Ideal Multiplexer benchmark, and hence can be modelled as an Ideal Multiplexer for a wide range of traffic parameters, with increasing accuracy as the number of stations becomes higher.

- The overall distribution of the generated traffic load interacts with the F-CPR's stop and wait nature between messages, to significantly impact global utilisation performance. Thus if only a small fraction of the active stations generates a large proportion of the traffic load, global utilisation is significantly lower than if the load is more evenly distributed among the stations.
- A large number of active stations with very small messages and light individual station load may lead to signalling channel congestion collapse due to collision of requests, particularly when the inter-station correlation is high. Even with the aid of F-CPR's contention-free "piggybacked" bandwidth reservation feature, such a signalling channel collapse is found to possibly lead to MAC protocol deadlock.

- Development of a suite of disaster scenario (deadlock) models and their analysis by means of a discrete-time Markov chain technique, and the introduction of the comparison between practical and theoretical stability (Chapter 4). In Chapter 6, we provide a detailed set of conditions which are shown to lead to practical instability and deadlock, and which depend on such factors as: signalling channel error probability, signalling traffic profile, and contention resolution algorithm properties.

- Proposing three new signalling channel capacity allocation schemes, with a view to extending the usable region of the F-CPR protocol by avoiding deadlock under as wide a range of conditions as possible (Chapter 4).

- Development of a new multi-priority scheduling scheme applied at the Head End central controller, based on a pre-emptive queueing principle (Chapter 5). Investigation of the impacts of intra- and inter-station correlation on the performance of existing and new multi-priority schemes, yielding important counterintuitive findings (Chapter 6):
  - Under extreme inter-station correlation, the average access delay of the high priority messages may significantly decrease with higher load, due to a combined positive effect comprised of: (i) the non-discriminatory nature of the contention-based signalling channel at lower loads, and (ii) the simultaneous preferential treatment of, and increased "piggybacked" bandwidth reservation usage rate by, the high priority traffic with increasing load.
  - The degree to which the high priority traffic delay decreases, depends on the level of intra-station correlation of the individual stations' streams, since high intra-station correlation enables F-CPR's contention-free "piggybacked" bandwidth reservation feature to be used more frequently.

- Proposing a new mechanism for the generation of multi-priority traffic from an (unprioritised) existing trace, based on the principle of having as many station subsets, with different sized populations, as there are priority levels in the system (Chapter 5).

2. Characterising Real Traffic

2.1 Importance of Using Self-Similar Traffic in Protocol Testing

In this chapter we explain why it is important to use traffic with self-similar properties, be it measured or generated by a model, in protocol testing and performance analysis. We also provide insight into the process of real traffic characterisation through a literature survey of state-of-the art models, all of which attempt to capture the inherent self-similarity in today's data network traffic. The last part of the chapter is devoted to the process of measuring and then characterising an actual traffic trace taken from the University's Ethernet Local Area Network (LAN); this trace is later used as our self-similar traffic stream in testing a particular Hybrid Fibre/Coax (HFC) access network protocol.

The strongest arguments for the use of self-similar traffic in testing are that: (i) it has been shown that a large proportion of traffic streams carried on modern data networks of today are underpinned by this property; and, (ii) self-similarity in a traffic stream invalidates many of the classic assumptions about correlation structure, and imposes a regime of memory, or, long range dependence (LRD) on the stream. Point (i) carries a lot of weight, since for accurate and effective design, we must strive to make the protocol usable under those conditions which capture the essence of the real-world traffic flow. It seems that in the past there has been a desire instead to use only hypothetical traffic, which as it happens may be described by "*nice analytically tractable*" mathematical models with classic parameters such as mean and short-term variance. The second argument for the use of self-similar traffic, outlined in point (ii), has at its roots the fact that self-similar traffic causes very different queueing behaviour, often leading to worse delay and throughput performance than memoryless traffic. In addition, the self-similarity often explores the limits and even some peculiar artefacts of the protocol being studied, as we shall later see in Chapter 6.

Having established the need for using self-similar traffic in protocol testing, we now turn to the next logical question, of which model is most suitable for synthesising the required traces. It is our opinion, after analysing the various state-of-the-art self-similar traffic models which are surveyed in Section 2.3, that there currently exists no model which can feed a set of certain traffic characteristics into a "conceptual black box" and obtain from this black box an accurate prediction of queueing performance. Therefore, it was decided to pursue an alternative source of a self-similar traffic trace - a measurement of live traffic crossing the University's LAN, as will be detailed in Section 2.6.

A measured trace consisting of real traffic also amounts to a model. However, it is a model which has as many parameters as there are trace readings, unlike a typical seven or eight parameter model of self-similar traffic. That is, the measured trace captures all possible traffic descriptors since it contains the traffic itself. The only problem with the "real trace model" is that the traffic could be wrong! Wrong in the sense that it might be totally unrepresentative of the traffic which will eventually load the protocol under study, in real operation. Unfortunately, this is going to remain a problem which can at best be minimised and not eliminated, since one cannot capture the traffic of a system which has not been built yet. The best

engineers can do to minimise the problem of unavailable accurate/realistic traffic measurements, is to build extensive, complicated prototypes.

However, careful choice of factors such as the type of end user equipment, applications being run on that equipment, and underlying network infrastructure, for which the measurements are made, can yield a traffic trace which can be considered, with confidence, as realistic for the system/protocol under study. Once we are confident of having a realistic trace, the measurement method has two distinct advantages over the use of a traffic model, which is why it is becoming increasingly popular in the network design paradigm. Namely, apart from being able to capture the full suite of traffic behaviour and characteristics, a measured traffic trace can be directly related to end user actions or application usage at certain instants, and thus give more intuition into protocol performance under such conditions.

To conclude, it does need to be said however, that a measured trace does have the significant limitation of being recorded in a static or “as is” manner. This means for example, that if a trace was recorded, and it did not “strain” the protocol under study with enough traffic load, it would be impossible to dynamically change some sort of simple parameter in order to increase this load. Instead, either a new trace with the desired load would need to be recorded (often very difficult to achieve), or some sort of manipulation of the trace would need to take place. We opted for the latter solution, the details of which are explained in Section 2.6.

2.2 Introduction and Definitions

It was a common belief that detailed modelling of connectionless traffic was practically useless because such traffic consists of different components varying both in time and from network to network [NORR 95]. That is why the discovery by Leland's group in Bellcore, that the multi-timescale burstiness of Local Area Network (LAN) traffic could be characterised by the simple notion of self-similarity, was so significant. Hundreds of millions of packets of aggregated Ethernet traffic were observed on several Ethernet LANs at the Bellcore Morristown Research Centre [LELA 91, LELA 94]. In another recent study, an analysis of a few million observed frame data obtained from the output of a Variable-Bit-Rate (VBR) video source, was also carried out [BERA 94]. Studies by Meier-Hellstern et al. [MEIE 91] have found that like the behaviour of connectionless Ethernet and video traffic, the ISDN D-channel packet data from an office automation environment exhibited strong features of self-similarity.

All of these studies show packet traffic to be statistically self-similar ("fractal"). The principal signature of self-similar traffic is its "burstiness" over an extremely wide range of time scales: traffic "spikes" ride on longer-term ripples, that in turn ride on still longer term "swells" - [LELA 94]. Figure 2.1, based on a similar figure from [LELA 94] illustrates this behaviour on the right side, while on the left it shows the increasingly white-noise like appearance of traffic generated by a Poisson model, as it is viewed on larger time-scales. The boxed sections of each of the figure’s graphs are “magnified” on a finer time scale, in the graph immediately below. Starting from the 0.01 s timescale resolution and moving upwards, note that the graphs in the right side of the figure remain as bursty as those with finer resolutions. However, as is

pointed out in [LELA 94], real network systems cannot display pure self-similarity extended out to an infinitely coarse (i.e. large) resolution scale. Ultimately, some physical bounds begin to be seen, and for example, in Figure 2.1, this may be interpreted as the emergence of some sort of a daily network usage cycle in the topmost right-hand graph, where there are distinct zones of high, low and medium activity.

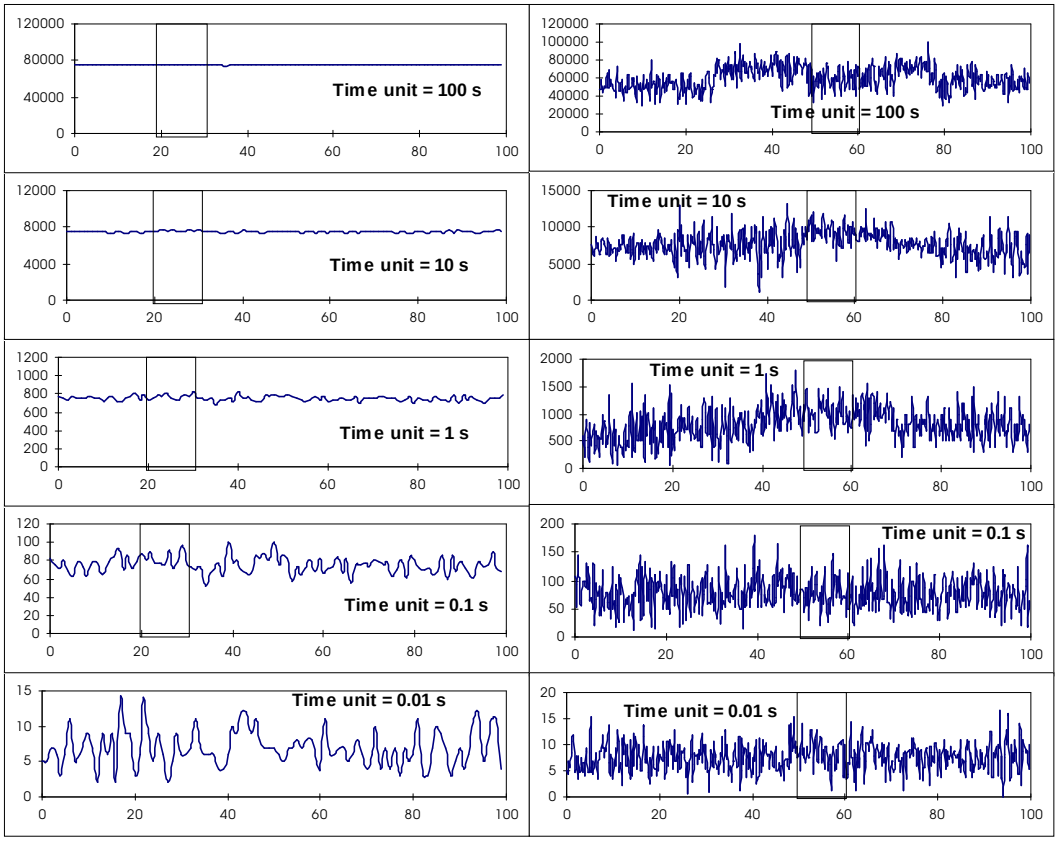


Figure 2.1: An Illustration of What Self-Similar Traffic Is

It is important to precisely define what is meant by ***exact*** / ***asymptotic*** second-order self-similar traffic. There is a number of published definitions, but using those from [LELA 94], we let $X = (X_t: \quad t = 0, 1, 2...)$ be a *covariance stationary* stochastic process, with mean μ, variance σ^2 and autocorrelation function $r(k), \quad k \geq 0$. Then for each m = 1, 2, 3... we let $X^{(m)} = (X_k^{(m)}: \quad k = 1, 2, 3...)$ denote the new "aggregated" time series obtained by averaging the original series X over non-overlapping blocks of size m. So for each m, $X^{(m)}$ is given by $X_k^{(m)} = (X_{km} + ... + X_{km-(m-1)}) / m, \quad k \geq 1$.

- The process X is called ***exactly second-order self-similar*** with self-similarity parameter $H = 1 - \beta / 2$, $(0 < \beta < 1)$ if for all m = 1, 2, 3..., the variance of the aggregated process is equal to,

$$\text{var}(X^{(m)}) = \sigma^2 m^{-\beta} \qquad (2.1)$$

  and the autocorrelation function of the aggregated process is equal to,

$$r^{(m)}(k) = r(k), \quad k \geq 0 \tag{2.2}$$

- X is ***asymptotically second-order self-similar*** with self-similarity parameter $H = 1 - \beta / 2$, if for all k large enough,

$$r^{(m)}(k) \to r(k), \quad as \quad m \to \infty \tag{2.3}$$

Therefore, X is exactly or asymptotically second-order self-similar if the corresponding aggregated processes $X^{(m)}$ have the same autocorrelation functions as X, or have autocorrelation functions which become indistinguishable from that of X in the limit.

It should be mentioned that [TSYB 96] states that the exactly self-similar definitions as presented in equations (2.1) and (2.2) from [LELA 94] have some level of redundancy, and are not suitable for extension to asymptotically self-similar processes. [TSYB 96] expresses the self-similarity definitions in the following way:

- Process X is called ***exactly second-order self-similar*** with parameter $H = 1 - \beta / 2$, if for all k = 1, 2, 3..., its autocorrelation function is,

$$r(k) = \frac{1}{2}\left[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}\right] \equiv g(k) \tag{2.4}$$

- Process X is ***asymptotically second-order self-similar*** with parameter $H = 1 - \beta / 2$, if for all k = 1, 2, 3..., the autocorrelation function of the aggregated process approaches *g(k)* as the aggregation level is increased,

$$r^{(m)}(k) \to g(k), \quad as \quad m \to \infty \tag{2.5}$$

The latter pair of definitions give more insight into the nature of the autocorrelation function of self-similar processes, because they show that, in the limit for large values of lag, *k*, *r(k)* decays *hyperbolically* rather than *exponentially*, as for traditional Markovian traffic models.

Namely, for exactly self-similar processes and large *k*,

$$r(k) \sim H(2H-1)k^{-\beta} \tag{2.6}$$

and for asymptotically self-similar processes,

$$r(k) \sim ck^{-\beta} \quad \textit{with some } \mathbf{c} \textit{ not necessarily equal to } \mathbf{H(2H-1)} \tag{2.7}$$

Self-similarity thus manifests itself in a number of mathematically equivalent ways: (i) the variance of the mean of an aggregated sample (of the original traffic process) decreases more slowly than the reciprocal of the sample size (slowly decaying variances as in equation (2.1)); (ii) the spectral density obeys a power-law near the origin; (iii) a non-summable autocorrelation function ($\sum_k r(k) = \infty$, indicating long range dependence), and (iv) an Index of Dispersion of Counts (IDC) that increases monotonically with the time interval length [LELA 94].

2.2.1 Behaviour of Multiplexed Self-Similar Traffic Streams

The work presented in [TSYB 96] has two statements which may be particularly useful in applications where data traffic from many streams is merging into one, as for example in an ATM switch or a data multiplexer:

Statement One

Adding two independent, asymptotically second-order self-similar traffic processes X' and X'', with parameters H' and H'' respectively, will result in another asymptotically second-order self-similar traffic process $(X' + X'')$, with a parameter equal to $H = max(H', H'')$.
$\Rightarrow$ *Merging two asymptotically second-order self similar data streams produces an asymptotically self-similar data stream.*

Statement Two

Assume that the independent processes X' and X'' are exactly second-order self-similar, with Hurst parameters H' and H'' respectively. Now, if $H' = H'' = H$, then the resulting process $(X' + X'')$ is also exactly second-order self-similar with parameter H. If however, $H' \neq H''$, then $(X' + X'')$ is only asymptotically second-order self-similar with $H = max(H', H'')$.
$\Rightarrow$ *Merging two exactly second-order self similar data streams may produce either an asymptotically self-similar data stream (if the Hs are unequal), or an exactly second-order data stream (if the two Hs are equal).*

2.3 Literature Survey of Models for Self-Similar Traffic

It was suggested by Leland et al. that it is very problematic to distinguish, for finite sample sizes (as they invariably are in practice), whether the asymptotic relationships for variance and autocorrelation, reflect true self-similarity, or are just an unavoidable artefact of finite data sets. The authors of [LELA 94] go on to describe three possible ways of modelling self-similar traffic.

The first two of these are formal mathematical models that yield "elegant representations of the self-similarity phenomenon": *fractional Gaussian noise*, first introduced in [MAND 68], and the class of

fractional autoregressive integrated moving-average (ARIMA) processes. The former model is exactly second-order self-similar, but only has three parameters (mean, variance and Hurst parameter) and thus a very rigid correlation structure, not capable of capturing the wide range of low-lag correlations encountered in practice. On the other hand *fractional ARIMA* models are a natural generalisation of the widely used class of Box-Jenkins models [BOX 76] with one of the parameters having non-integer values. These processes were introduced by Granger and Joyeux [GRAN 80] and Hosking [HOSK 81] who showed that they were asymptotically second-order self-similar, and much more flexible than fractional Gaussian noise in that they are able to simultaneously model both short- and long-term dependent behaviour.

The third method mentioned in [LELA 94] consists of a construction of self-similar processes, based on aggregating many simple renewal reward processes which exhibit inter-renewal times with infinite variances (for example a Pareto distribution). This way of modelling fractal traffic was introduced initially by Mandelbrot [MAND 69] and was subsequently extended by Taqqu and Levy [TAQQ 86]. Unlike the two formal mathematical models, which provide an elegant representation without any physical explanation of self-similarity, this aggregated renewal reward process may be intuitively thought of as a *packet train model* (as detailed in [JAIN 86]). The "rewards" take on the values 1 or 0 (active/inactive periods), while the inter-renewal intervals exhibit, using Mandelbrot's terminology, the *infinite variance syndrome*. This can be thought of as a single source model where the value 1/0 during a renewal interval indicates an active/inactive period, during which a source sends 1/0 unit(s) of information, every time unit. This third model at least provides some insight into the underlying causes of self-similar behaviour.

The fractal behaviour of traffic [MAND 65, MAND 83] from diverse applications is very different both from conventional telephone traffic and from the models which were considered for packet traffic up until now (for example Poisson, Batch-Poisson, Markov-Modulated Poisson Process, Fluid Flow models etc.). In fact, it was found during the Bellcore studies reported in [LELA 94], that contrary to common beliefs that multiplexing traffic streams tends to produce smoothed-out aggregate traffic with reduced burstiness, aggregating self-similar traffic streams actually intensifies burstiness rather than diminishing it. The studies showed that the degree of self-similarity of the Ethernet traffic increased as the utilisation of the Ethernet increased.

The Leland et al. Bellcore studies also pointed out the problem with use of the *peak-to-mean ratio* as a measure of traffic burstiness, in the presence of self-similar traffic. It is pointed out, that in fact any peak-to-mean ratio is possible, depending on the length of the measurement interval.

As pointed out in [DIAM 96], fractal behaviour raises questions concerning the performance of statistical multiplexers loaded by traffic of this nature. In practice, ATM switches need to make decisions (in real time) about how many sources can be multiplexed together without causing too high a cell loss rate, either due to buffer overflow or excessive delay discarding. Many investigations have been conducted into how the self-similar behaviour of data streams, from very diverse applications, will impact the performance of future ATM networks.

In [CHEN 95a], Chen et al. developed simulation models in order to predict the impact of self-similar traffic on the cell loss and cell delay performance of an ATM switch. The simulated traffic streams had two parameters: the mean packet arrival rate (network utilisation), and the Hurst parameter H [HURS 51]. The Hurst parameter, $0.5 < H < 1.0$ expresses the degree of self-similarity in a traffic stream: pure Poissonian traffic has $H = 0.5$, while for $H > 0.5$, the higher H is, the more "burstiness" over wider time scales will be exhibited by the traffic stream. It was found by the authors, that the self-similarity of the traffic had a potentially serious effect on the switch performance. The simulation models proposed in [CHEN 95a] predict both cell loss and cell delay to be orders of magnitude higher than would be predicted by more traditional traffic models (e.g. a simple Batch Poisson process).

In [DIAM 96], Diamond and Alfa describe an analytic traffic model based on renewal processes, the discrete-time HYPER/D/1 queue, which is less complex and less computationally intensive than the traffic generation scheme proposed in [CHEN 95a], yet retains the same characteristic of describing the traffic streams with two parameters. Although Diamond and Alfa's work illustrates that it is possible to develop useful and relatively simple analytical queueing models which capture self-similar behaviour, it points out that it is based on an unrealistic assumption of an infinite buffer. This has the effect of underestimating the utilisation of the queue, at times of nearly full buffer loading. It is pointed out in [DIAM 96] that matrix analytic methods can also be used to analyse finite buffer models (e.g. the HYPER/D/1/b queue), however the computational requirements significantly increase for realistic buffer sizes (i.e. large b).

In related publications about analytical results for queues with self-similar input traffic, Duffield and O'Connel, Likhanov et al., and Norros have also made important contributions. In [DUFF 93], Duffield and O'Connel obtain the asymptotic form of the decay of tail probability for queue length in models with self-similar traffic input.

Likhanov et al. [LIKH 95] present an analytical study of an ATM buffer driven with self-similar traffic, and show that the probability of buffer occupancy decreases with the buffer size algebraically, and not exponentially as in traditional Markovian traffic models. The approach taken in this work is quite novel, in that the model is a superposition of an infinite number of constant-rate on-off sources with Pareto distributed on periods. The G/D/1 queueing model, (resulting from a finite buffer, deterministic service ATM switch) is then mapped into an equivalent and easily treatable M/G/1 queueing system, representing the Poisson arrivals of activity-bursts of the Pareto on-off sources. The queue length of the M/G/1 system is then obtained and gives the stationary distribution of the number of currently active sources (i.e. those which haven't yet been "served" according to their Pareto service demands).

The work of Tsybakov and Georganas [TSYB 96], partially based on earlier research in the above mentioned [LIKH 95], considers a large class of asymptotically self-similar processes, in order to obtain a suitable model of input to an ATM queue. By using the class of models first proposed by Cox [COX 84], where Poisson-distributed arriving bursts have Pareto-distributed active periods, the authors of [TSYB 96] theoretically derive lower bounds for buffer overflow and cell loss probabilities, as well as the stationary distribution of the cell delay in an infinite ATM buffer. It is shown that, as a direct consequence of self-

similar input, and in contrast to those of traditional Markov models which decayed exponentially, the buffer overflow and cell loss probabilities cannot decay faster than hyperbolically with the buffer size. It is stated in [TSYB 96] that these two theoretical findings *"firmly establish that ATM buffers, designed under traditional (Markovian) traffic modelling and analysis, should be increased significantly, in order to provide adequate QoS to traffic exhibiting self-similar features"*.

The work in [NEAM 95] attempts to fit the "M/Pareto" model proposed by Likhanov et al. [LIKH 95], to real traffic, measured from a Wide Area Network (WAN). Both the original traffic and the traffic generated by the model-fit are passed through a finite buffer single server queue (SSQ), and the resulting proportion of cell loss is used as the comparison yardstick. The main goal of this work was to investigate the conjecture that it is enough to fit (i) the Hurst parameter (via the time-variance curve), (ii) the mean arrival rate, and (iii) the variance of the marginal distribution, in order to obtain a simplified model of the real traffic which predicts performance accurately.

The [NEAM 95] paper shows that the "M/Pareto" model succeeds in fitting the real traffic marginal distribution quite well, especially in comparison with two other candidate models - the Fractional Brownian Motion and the Sum of Two AR(1) processes (details about the latter model are given in [ADDI 95]). In addition, the fitting of the variance-time curve is good for the "M/Pareto" model. Despite this, the model doesn't do very well at estimating the probability of loss, which is the criterion by which traffic models are ultimately judged. A significant observation also made in this research is that the two Gaussian models mentioned above (F.B. Motion and AR(1)), one with a Hurst parameter $H = 0.5$, and the other with $H > 0.5$, can have very similar queuing performance. This is observed although the traffic generated by the Fractal Brownian Motion model, with $H > 0.5$, is asymptotically self-similar and would be expected to yield worse queueing performance.

One-dimensional chaotic maps, in which the evolution of a state variable over discrete time is described by a non-linear transformation, were used in the work by Erramilli and Pruthi [ERRA 95] which attempted to tie in heavy-tailed behaviour of ON/OFF sources and self-similarity in traffic. It was found that long range dependent traces can be generated by sources which have heavy-tailed OFF behaviour (sustained inactivity periods), regardless of the nature of the ON period, as measured by variance-time plots and power spectra. Given a source with a heavy-tailed OFF period, the nature of the ON period does however produce significant differences in queueing behaviour: sources with heavy-tailed ON periods generate queue length distributions that decay as power laws, while light-tailed ON period sources produce exponentially decaying distributions.

Erramilli and Pruthi state that the above paradox of traffic traces with similar second-order statistics, yet dramatically differing queueing behaviour, may be explained by differences in higher-order statistics. This hypothesis is supported by considering that aggregates of heavy-tailed ON/OFF sources, regardless of the nature of the ON period, converge to Fractal Brownian Motion (FBM) and produce Weibullian "stretched exponential" queueing behaviour consistent with that of FBM models. Of course aggregation of independent sources has the effect of diminishing the impact of higher order statistics.

In [ERRA 95] it is also pointed out that the physical basis of observed self-similar behaviour is valuable in providing insights into the performance impact of long range dependence. The authors go on to postulate that Ethernet sources fit into the category of heavy ON/OFF source behaviour. Single sources of this type exhibit wildly volatile queueing behaviour and may need peak rate allocation in order to avoid serious performance degradation; aggregation causes fairly quick convergence to FBM, yielding so called "exactly self-similar" traces. On the other hand, Signalling System 7 (SS7) sources appear to fit into the category of heavy OFF/ light ON behaviour, with far less resource-demanding queueing behaviour. Aggregation of sources of this type (across sources and in time) leads to a markedly slower convergence to FBM, yielding "asymptotically self-similar" traffic traces. A final concluding remark was that a traffic stream consisting of many limited bursts, will at high enough utilisation be indistinguishable from a stream generated by extended bursts.

In [LIU 95], Liu et al. model the traffic arrival process in an ATM LAN, with some likeness to the work in [NEAM '95] and [LIKH '95]. Liu et al. report that, based on extensive Ethernet and ATM LAN measurements, it was found that the arrival intervals of cell traffic were neither purely Pareto nor Negative Exponentially distributed. Therefore a hybrid distribution containing both the Pareto and Negative Exponential features was proposed, and successfully fitted to the measured probability distribution of cell arrivals. It was stated that the Pareto component represents the clustering of events over short time scales, and a Poisson component represents the independence of clusters over long time scales.

In summary, the two approaches which were tried with the original Likhanov et al. "M/Pareto" model are as follows:

1. Representing the ATM switch as an equivalent infinite buffer M/G/1 queue, where the Poisson distributed arrivals of constant-rate bursts are served according to a Pareto distribution, one-burst-at-a-time. This approach was taken in [LIKH 95] in order to arrive at an analytic expression for the probability distribution of the queue length (i.e. number of sources waiting in the queue, since they are in an unfinished burst state).

2. Discretising time into intervals, and modelling a Poisson arrival process of bursts with Pareto-distributed lengths. Each burst generates cells at a constant rate during its lifetime, and one or more bursts may overlap across some intervals, with the ATM switch serving cells at a deterministic rate. This approach, equivalent to the one in 1. above, was taken in [NEAM 95] because it is easy to simulate.

3. Rather than model the Poisson distributed arrivals of bursts, each of which could be identified as belonging to a given source, [LIU 95] opted for modelling the individual cell inter-arrival times by a Hybrid (Pareto / Poisson) distribution.

This summary has identified the obvious need to further investigate the promising "M/Pareto" model, especially by taking the approach whereby one models Poisson arrivals of constant-rate bursts, the lengths

of which have a Hybrid (Pareto / Poisson) distribution. As a side note, the expression characterising this particular hybrid distribution is identical to the Bond function. Namely, in earlier work on theoretical studies of noise in digital communication channels [BOND 87], Bond develops a mathematical description of burst noise, and describes two parameters: the *burstiness*, being the tendency of the errors to cluster together; and the *noise rate*, which is the rate at which errors occur. On the way to formulating an expression characterising burst noise, one of the resulting functions (B(t)) takes the form of a product of Pareto and Negative Exponential (i.e. Poisson arrivals) complementary cumulative distributions.

In [NORR 94] Norros first publishes the *fractional Brownian traffic* model, and obtains a relation coupling the storage requirement, achievable utilisation and the output rate for a storage model with self-similar (*fractional Brownian traffic)* input. In more recent work [NORR 95], Norros defines the concept of *"Free Traffic"* as "an ideal notion for what the traffic would be if the network resources were unlimited". This work then makes the conjecture (subject to further work in the future) that the fractional Brownian traffic model is a generally applicable model for Free Traffic, subject to the condition that it is aggregated from a sufficiently large number of independent sources, whose peak rates are substantially lower than the combined mean rate.

In their work published in [GARR 94], Garrett and Willinger analyse in detail a two hour long empirical sample of VBR video, derived from an action movie ("Star Wars"). As in the research referred to earlier, the video frames (and slices) exhibit strong self-similarity features such as a long-range dependent time correlation structure, and a heavy-tailed marginal distribution of the information content per time interval. Long-range dependence (also "persistence" or the "Hurst effect") is defined as "the phenomenon of observations of an empirical record being significantly correlated to observations that are far removed in time". These findings are combined in [GARR 94] to yield a new non-Markovian source model for VBR video. The model captures the heavy-tailed marginal distribution by use of a hybrid Gamma/Pareto distribution, while the autocorrelation function with long-range dependence is attained by using a fractional autoregressive integrated moving average fARIMA(0,d,0) process, where the zeros indicate that there are no autoregressive (AR) and moving average (MA) parameters specified.

An extension to the original fARIMA model is proposed in [QURE 95], and is aptly named the extended fractional ARIMA, or e-fARIMA, model. [QURE 95] explains that the main contribution of this new model is chiefly in its ability to accurately characterise and predict the behaviour of VBR traffic; e-fARIMA can reportedly account not only for the long and short range dependencies within the VBR stream, but also for the quasi-periodic correlations typical of VBR traffic. The paper also formulates a linear minimum mean square error (lmmse) predictor for e-fARIMA, on the basis of which one can determine the queue length in a constant service rate, single-server ATM queue fed by e-fARIMA traffic. Simulation results are presented highlighting that the model is effective and accurate, but [QURE 95] also explains that the price of this accuracy is a much larger number of parameters needed than for other self-similar traffic models. For example, a simple self-similar FBM model has only three parameters, as compared to at least eleven parameters needed in e-fARIMA (the exact number of parameters depends on the order of the underlying fARIMA process).

A fast simulation approach to simulate self-similar traffic was developed by Huang et al. in [HUAN 95], in order to overcome the problem whereby predicted performance based on steady-state theoretical and simulation methods may be overly pessimistic for practical applications. This postulation arises from the notion that, while the self-similar property captures the burstiness of traffic at all time scales, realistic ATM networks are expected to have a limiting time scale. It was also pointed out in [HUAN 95], that theoretical approaches for obtaining transient queueing behaviour and distributions for small buffer sizes become quickly intractable. Therefore, [HUAN 95] presents a fast simulation approach based on importance sampling (IS) and Hosking's method [HOSK 84], and simulates the transient queueing behaviour of the discrete-time self-similar arrival process termed fractional Gaussian noise (FGN).

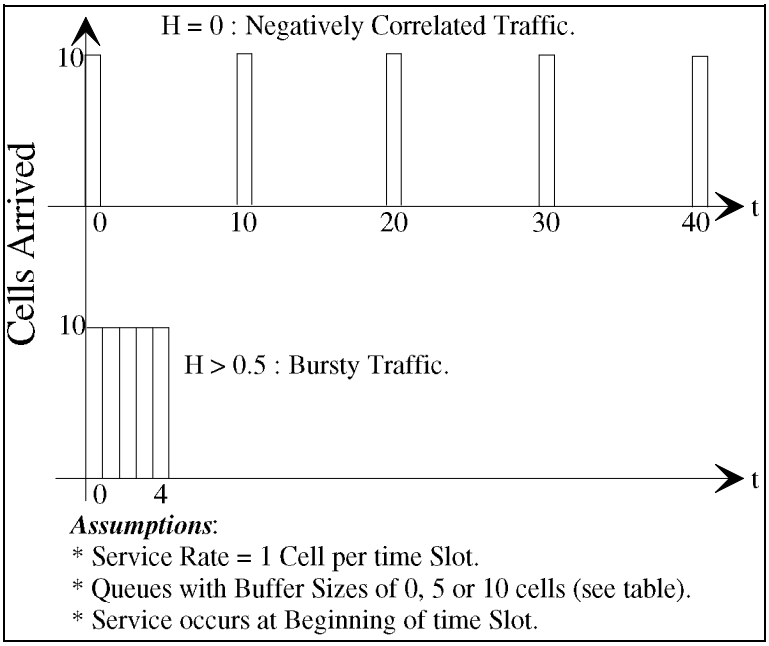
The advantage of the approach described in [HUAN 95] over others for the synthetic generation of self-similar traffic traces (e.g. Mandelbrot's fast FGN generator [MAND 71] and non-linear chaotic maps [ERRA 90]) lies in the ability to provide a sufficiently large number of replications, thus allowing accurate statistics to be obtained. The [HUAN 95] paper also focused on the key ATM design issues of *buffering gain* (defined as the reduction in cell loss probability due to increasing buffer size) and *multiplexing gain* (defined as the reduction in cell loss probability due to statistical smoothing when multiple bursty sources are aggregated). The significant results observed by Huang et al. were: (i) the higher the burstiness, measured by the Hurst parameter H, the lower the buffering gain (as was just illustrated in the example from Figure 1.2); (ii) compared with "traditional", or, short range dependent (SRD) traffic models, self-similar models show smaller buffering gains; (iii) contrary to the famous Bellcore Ethernet studies' results, increased multiplexing gains were obtained with more bursty self-similar traffic (higher Hurst parameter value); (iv) in multiplexing two heterogeneous self-similar sources, the steady-state queueing behaviour will be dominated by the burstier one, so that, when a process possesses both long and short range dependence structures (e.g. as in the case of traffic from an fARIMA model), the steady state will only reflect the contribution of long range dependence.

2.4 Buffering Gain Issues for Self-Similar Traffic

[LAVA 96] looks at the sensitivity of large and small buffers, when loaded by self-similar, positively correlated traffic ($H > 0.5$) and negatively correlated traffic ($H < 0.5$). It is found that larger buffers are more sensitive (suffer comparatively worse loss performance) to traffic with high values of H. This is intuitive because, larger buffers have "longer memories" and are therefore more strongly affected by long range dependent, strongly correlated traffic (traffic with large values of H). If we focus our attention on the example presented in Figure 2.2 (overleaf), it is clear from the table that a *zero buffer queue*, being totally memoryless, suffers an equally bad cell loss proportion (0.9) whether the traffic is periodic and deterministic ($H = 0$), or whether it is extremely bursty (large H).

From tabulated results for the other two queues, with buffer sizes of 5 and 10 respectively, the main observation we can make is that even in this simple example, the larger the buffer size of a queue, the greater the disparity in loss performance for traffic with large and small values of H, respectively. Namely,

we observe the two types of traffic at loss levels of *{0.9, 0.9, buffer = 0}*, then *{0.8, 0.4, buffer = 5}* and finally at *{0.7, 0, buffer = 10}*.



Traffic Type	*Buffer Size (cells)*	*Cell Loss Ratio*
H = 0	0	0.9 (45/50)
H > 0.5	0	0.9 (45/50)
H = 0	5	0.4 (20/50)
H > 0.5	5	0.8 (40/50)
H = 0	10	0.0 (0/50)
H > 0.5	10	0.7 (35/50)

Figure 2.2: The Effect of Self-Similarity, as measured by *H*, on Buffering Gain

An alternative way of describing this phenomenon is that the higher the burstiness, measured by the Hurst parameter *H*, the lower the *buffering gain* (defined as the reduction in cell loss probability due to increasing buffer size) [HUAN 95]. Of course, the more obvious result is that with increasing buffer size, better performance for both types of traffic will be apparent, due to more storage space being available. However, as explained, it is the buffering gain which will determine exactly by how much performance will be bettered.

2.5 Is Multiplexing Gain Possible for Self-Similar Traffic?

The possibility of network efficiency being improved in large core networks of the future is explored in [ADDI 97]. *Network efficiency* may be described as the level of utilisation achievable at a given quality of service. As mentioned in [HUAN 95], we define multiplexing gain as the reduction in cell loss probability due to statistical smoothing when multiple (bursty) sources are aggregated. Or, conversely, it is the

increase in network utilisation given a fixed cell loss probability, achievable due to the statistical smoothing of traffic.

[ADDI 97] also defines two extreme positions about call acceptance strategies and network control, that have appeared over the years in the literature studying broadband networks:

- **Fixed Bandwidth Approach** - virtually all traffic in the future B-ISDN should be viewed as a "fixed bandwidth" circuit. A fixed and dedicated bandwidth channel (i.e. Constant Bit Rate or CBR in ATM terminology) is allocated to any call/connection upon set-up, and is freed up upon call completion. This point of view used to be in ascendancy with international standards bodies according to [ADDI 97].

- **Adaptable/Internet Approach** - allow all calls/connections to use the available bandwidth and monitor the traffic levels, in order to avoid bad performance. Occasionally a call/connection might experience bad cell loss because the rest of the network became suddenly and unexpectedly busy. This point of view reflects today's design of the Internet, which does work well, yet currently provides no guarantees with regards to delay and data delivery.

Thus, if a suitable traffic model can be developed for the communication flows of the future, then each of these points of view can be thoroughly evaluated. The issue then remains, given that a multitude of recent studies have found traffic likely to be carried on the core network to be self-similar, of *"what multiplexing gains are achievable, as the volume of traffic carried by the core networks grows?"*.

The view of Addie in [ADDI 97] was that the answer to this question lies in the eventual behaviour of highly aggregated, large volumes of traffic which will flow in the core networks of the future. Namely, using a recent result (from [ADDI 96]) which suggests that network traffic will converge to a Gaussian character as more and more traffic is mixed together, [ADDI 97] hypothesises that the relatively simple Gaussian Fractional Brownian Motion (FBM) model, which also captures self-similarity properties, might be appropriate when large enough amounts of traffic are aggregated.

By means of numerical experimentation with a queuing model in a variety of possible situations (significant variation in the values for Hurst parameter (*H*), variance and loss probability), [ADDI 97] shows that if the assumptions of the paper hold (i.e. that the traffic can truly be modelled by a Gaussian FBM process), significant multiplexing gains will be achievable as the volume of traffic in the core network grows. The key finding is that for the proposed Gaussian FBM model, an *n*-fold increase in the number of homogenous sources causes corresponding *n*-fold increases in mean and variance, and no change in the Hurst parameter, *H*. Hence, although the aggregated traffic retains the same self-similarity level as before, the standard deviation to mean ratio reduces by a factor of $\sqrt{n}$. This reduction corresponds to a direct multiplexing gain benefit.

Importantly, and perhaps contrary to the findings in [LELA 94], it is also observed that larger multiplexing gains are experienced by traffic which is *more self-similar*. This observation lends weight to the theory that there is no reason why the self-similar property of traffic will make it difficult for future core networks to achieve solid multiplexing gains upon aggregation. In fact, from the results presented in [ADDI 97], the opposite is true: more self-similar traffic will increase the benefits of aggregation, through better relative multiplexing gain, than less correlated traffic.

It is also pointed out that these conclusions strongly rely *on "sufficient traffic being aggregated together for the Gaussian FBM model to be applicable"*; and, that there is still insufficient knowledge regarding how large the level of aggregation will have to be (i.e. the central core of the network), in order for this to be the case. However, if the core network traffic flows of the future *can* be accurately modelled by the Gaussian FBM model presented in [ADDI 97], and the parameters *can* be estimated accurately, then the experiments described in this paper show that network performance can be maintained at acceptable levels, by allowing all connections to use the common bandwidth and then keeping the network traffic within a parameter space in which performance is acceptable (as governed by the *Adaptable/Internet* approach described above).

2.6 "Real" Traffic Files: Measurement Details and Trace Properties

2.6.1 Measurement Set-Up

In the absence of actual Cable Modem traffic measurements, our "real" traffic is based on an Ethernet traffic trace (and subsets of this main trace) generated by several different campus workstations running typical applications. Some of today's standard office/university campus applications such as Web browser access to the Internet, file transfer (ftp) and electronic mail (e-mail), carried via Ethernet LANs today, will also run on Cable Modem HFC networks. As was mentioned previously, there is an added benefit in using measurements from the Ethernet, because it is emerging as the interface of choice between the Cable Modem and household PC, for virtually every manufacturer in the industry. We feel that these two facts justify our choice of the applications we measure, and the Ethernet network infrastructure we perform our measurements on.

In line with these two observations, we adopted the approach of constructing a large concatenated traffic file, with close to half a million seconds of Ethernet readings, taken periodically (1 second) at the Ethernet ***out*** interfaces of four different university UNIX workstations, one of which was a small server. Each reading records the number of Ethernet frames that has exited the ***out*** interface during the 1 second period. Since the HFC system simulations, which will be run with this recorded traffic, are based on time periods ("slots") which are between four and six orders of magnitude smaller than the 1 second measurement interval, it would be erroneous to simply use each reading as the number of Ethernet frames departing per slot. This would be a physically inaccurate use of the data, because the HFC system slots are so much

shorter in duration than one second (typically around four orders of magnitude) that the only two viable outcomes are *no slot-based frame departs* or *one slot-based frame begins to depart*.

Hence, our first step in interpreting the recorded data is to scale-down to the HFC system slot-level treating each zero reading as the absence of a frame departure during a slot, and each non-zero reading as the beginning of a frame departure during a slot (i.e. during this slot, a cable-modem receives an Ethernet frame from the PC for transmission to the Head-End).

The second step is to determine the size, in equivalent ATM cells of each of these slot-based outbound Ethernet frames (recall that for the purposes of the F-CPR protocol we study here, HFC system slots are based on the time duration to transmit a single ATM cell). Given that it was not easily possible to obtain the true frame size information, rather than assume an arbitrary frame size distribution, we use the original 1 second based reading (of the number of Ethernet frames) to *translate to the size in cells*, of the slot-based departing frame. For example, readings with 5 and 10 original Ethernet frames per second approximately represent a ratio of 1:2 in the volume of data per second, on the average, (assuming similar size distributions for the individual frames within these two readings). So in order to retain this ratio when the unit time becomes a single slot, we assign slot-based frame sizes also in the 1:2 ratio. The simplest mapping would be direct translation - assign frame sizes of 5 and 10 cells respectively.

We consider this interpretation of the raw readings to have merit, since an original reading with a certain amount of traffic per second, represents a given volume of outbound data per unit time; then, by assigning slot-based frame sizes proportional to this original volume, much of the information in the original traffic trace is retained in the scaling-down from 1 second to a single slot duration. In particular, the correlation structure (i.e. the short- and long-term *"memory"*) of the amount of work departing, is largely retained. Note that the chief assumption here, and the only significant weakness of this scaling-down approach, is that the frame size distribution (measured in Bytes per frame) for each original reading, X_n frames, is assumed not to vary.

That is we assume that, on average (in many trials), if reading X_n equals say, reading X_{n+100}, the respective volumes of data (measured in cells per frame) are also equal since the original frame sizes (measured in Bytes per frame) are identically distributed. Many of the original readings within the file are "0 frames" - this simply means that during the interval in question, there were no Ethernet frames at the output interface (translating also to no outbound frame at the slot-level).

The concatenation stems from the fact that about four equally sized files were serially joined one to the other, in order to form one large composite file. More details of this concatenation and the reasoning behind it is explained in the following section. Note that measurement of the Ethernet ***in*** interface was also possible, but due to the broadcast nature of the Ethernet medium, it would have been much more difficult to establish how much of the inbound traffic was actually intended for the given computer.

The relative values of load offered to the network by each of the four workstations, and the type of application(s) running on them is shown in Figure 2.3.

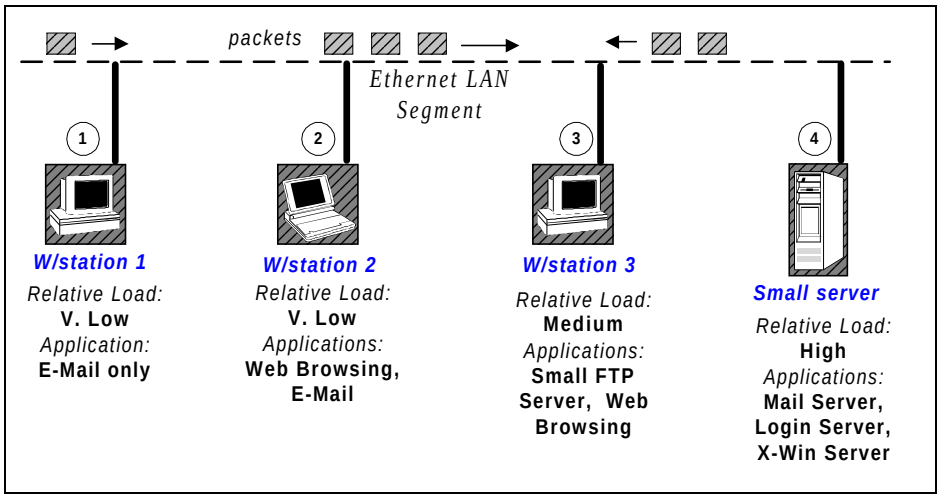


Figure 2.3: Measurement Network Topology

2.6.2 Run-time Trace Read Process

The idea behind creating a long composite concatenated traffic trace stemmed from the realisation that we would have to test the protocols and ideal multiplexer for system sizes in the order of 10 to 200 stations. The alternative to what we did would have been to individually collect equal-length traces for each station. This method has serious practical difficulties:

⇒ As the simulated system got large, (>20 stations) it would be very difficult logistically to collect enough unique station traces, while making sure that each was being used for its intended application set (e.g. just E-Mail, when the user wants to "surf the net").

⇒ If say a top of 20 different station traces were successfully collected, and it was desirable to test a 200 station system, then a 10:1 duplication ratio would necessarily have to arise, introducing unwanted cross-correlation effects.

⇒ Even if each station had a separate trace file to read from, it would need an active *file pointer* in order to perform the read. Manipulation of (>16) different and concurrently active file pointer structures in most programming languages is either unachievable or places such a burden on the processor that the simulation virtually cannot run.

On the other hand, our composite trace method focused on a single file which was sufficiently long to allow it to be dynamically subdivided in a mutually-exclusive manner (i.e. so stations always read non-overlapping data). By dynamically, we mean that the simulation decides at run-time, depending on which system size (in stations) the user has specified, what is the maximum number of samples that may be read by each individual station and where the inter-station boundaries are in the file, so that no overlap occurs.

In this way, we automatically run the simulation longer for systems with fewer stations, and vice-versa. The concept of this dynamic subdivision is illustrated in Figure 2.4. The primary advantage of this method is that many stations can be arbitrarily simulating the generic traffic characteristics of various simulated applications (depending on which region of the file they are reading) without duplicating actual numbers of other such stations.

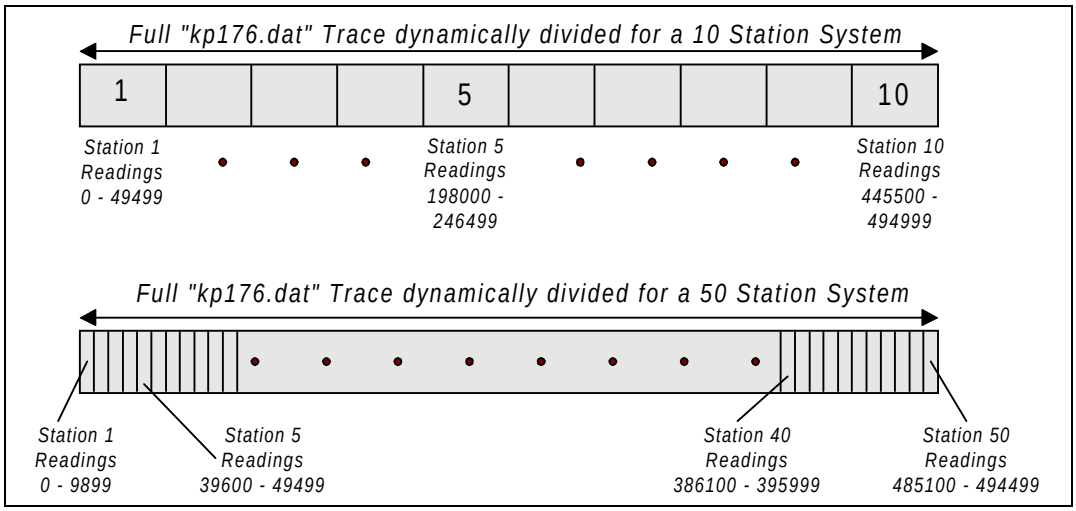


Figure 2.4: Dynamic Subdivision of Composite Trace at Run-Time

2.6.3 Controlling Trace Replay Speed

The value of any input traffic in protocol testing lies in the ability of the tester to vary the average arrival rate, or, in this case, the *replay speed* associated with the traffic trace. The fact that we could not decrease the periodic measurement interval between successive packet-size readings to a level of finer resolution than one second, is therefore largely irrelevant. In the replay speed manipulation process, we inserted uniformly distributed silence intervals, which we denote with L_I, where I = 1, 2, 3 ... represents the interval index. The average length of these inserted silence intervals is denoted by L, and they are uniformly distributed in the range of integers between $L/2$ and $3L/2$. That is,

$$L_I \sim U\left(\frac{L}{2}, \frac{3L}{2}\right), \quad for \quad all \quad I = 1, 2, 3... \tag{2.8}$$

By then increasing or decreasing the mean, L, we would be simulating a smaller or larger arrival rate, respectively, by varying the read frequency. This methodology allows us to fine tune the offered system load to a discrete time system with a service rate which is not easily changeable; and, at the same time, this permits us to largely retain the variance and autocorrelation characteristics of the original trace.

The latter point is of paramount importance, since we are seeking to test the true characteristics and behaviour of the trace, with as little “replay artefacts” as possible. On one hand, when testing the system performance with a particular trace, we would want ideally to deterministically sample readings from the trace, because it was recorded in such a manner. On the other hand, this would mean that, for every

individual replay speed, once the stations were randomly assigned starting timeslots at the beginning of the simulation, they would retain their read-times relative to each other for the whole length of the run. If we conceptually thought of time divided into intervals of length L (representing one read cycle), then during each such cycle both the order of station reads, and spacing between the reads would be exactly the same.

This outcome is highly undesirable because it creates a de-facto TDM system with false periodicity of transmission, equivalent to sampling each station after exactly L timeslots (c.f. a G.704 2 Mbit/s channelised TDMA structure with periodic repetition of 32 timeslots per frame). Signalling collisions are then made impossible (for the entire simulation run) for stations assigned different timeslots at initialisation. Far from being realistic, the performance results (delay and utilisation performance) would show highly pronounced simulation artefacts. Even for a given replay speed, key statistics of the overall system's input process such as correlation, variance and probability of signalling collisions would be extremely sensitive to the totally random assignment of stations' starting timeslots at initialisation. This is an unacceptable situation where at the same load, each different Random Number Generator seed may potentially yield wildly varying performance measures (in testing this method it was often found that higher load could yield smaller delay!).

To overcome this problem, we chose the compromise solution of a uniformly distributed L_I, as indicated previously by equation (2.8), and shown in Figure 2.5. The reason this approach is branded a compromise is because (a) for each simulation run L_I is not constant; however, the average of many samples is equal to a nominal value L, and (b) because L_I was uniformly distributed around L, we have not introduced any new distribution-related "skew" to the existing trace arrival process (as would have been the case for, say, an exponential distribution). Thus, in this way we are for the most part preserving the statistical properties of the recorded trace, while introducing a desired level of randomness in order to more realistically emulate the asynchronous and contention-based behaviour of stations on the shared HFC medium.

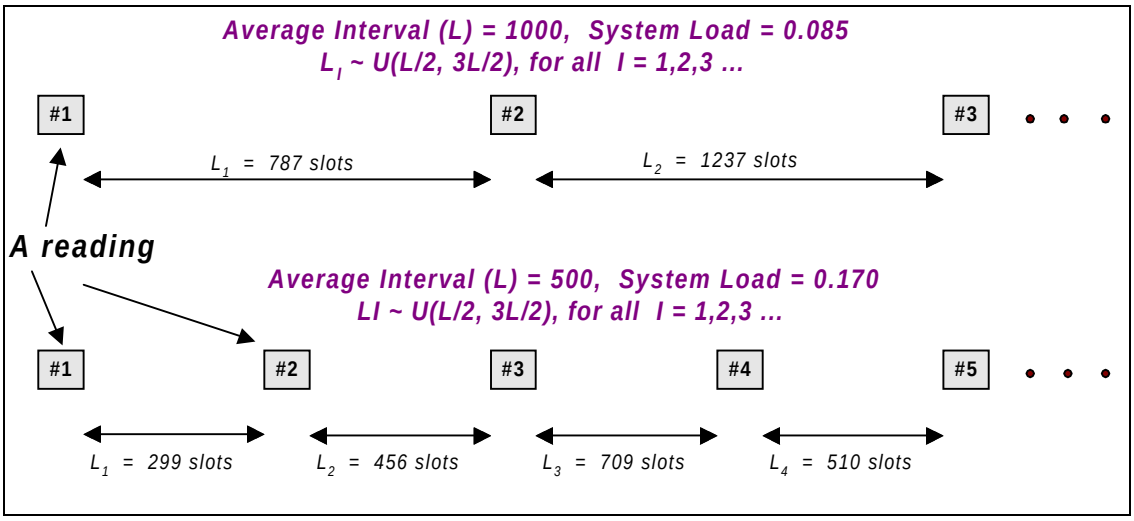


Figure 2.5: Replay Speed - Inserted Silence Intervals

2.6.4 Composite Trace Statistics

The entire composite trace, (for future reference, called *"kp176.dat"*) is pictured in Figure 2.6, with the vertical axis showing the interval's mean message size (already converted into a measurement of ATM cells) and the horizontal measurement giving the interval number (ranging from 0 to 99). Note that the huge trace size (~0.5 million readings) prevents a direct plot for each and every reading, and so in order to produce a meaningful graphical representation, we have had to subdivide the file into 100 intervals, each with 4549 readings; we then show the average message size in cells for each of the newly created intervals. This averaged aggregation does reduce the burstiness of the trace, but it is done for graphical presentation purposes, since it would be near impossible to go to a finer resolution with so many numbers. Figure 2.6 also shows the boundaries delineating each of the four workstations' individual files. It is interesting to note how quiet the E-Mail and Web browser computers are (Workstations 1 and 2), in relation to the network transmit activity of the server and the FTP workstation (Server and Workstation 3).

This arises due to the nature of the applications in question. E-mail usually consists of small text based files, and so requires virtually no network resources, save for very small and very rare bursts. Similarly, Web browsing in the current Internet environment has been shown to require only limited upstream bandwidth, due to the nature of "web surfing" which consists of very small and infrequent request packets (mouse clicks for HTTP/FTP/TELNET links) upstream, soon followed by a much larger and prolonged flow of downstream information packets (a file, a picture, a HTML page) [BISD 96b]. On the other hand, the workstations which act as FTP servers and X-windows servers will need to transmit information regularly and in much larger amounts, since it is they who are acting as the information source. These patterns are clearly evident in the four individual segments of our composite trace.

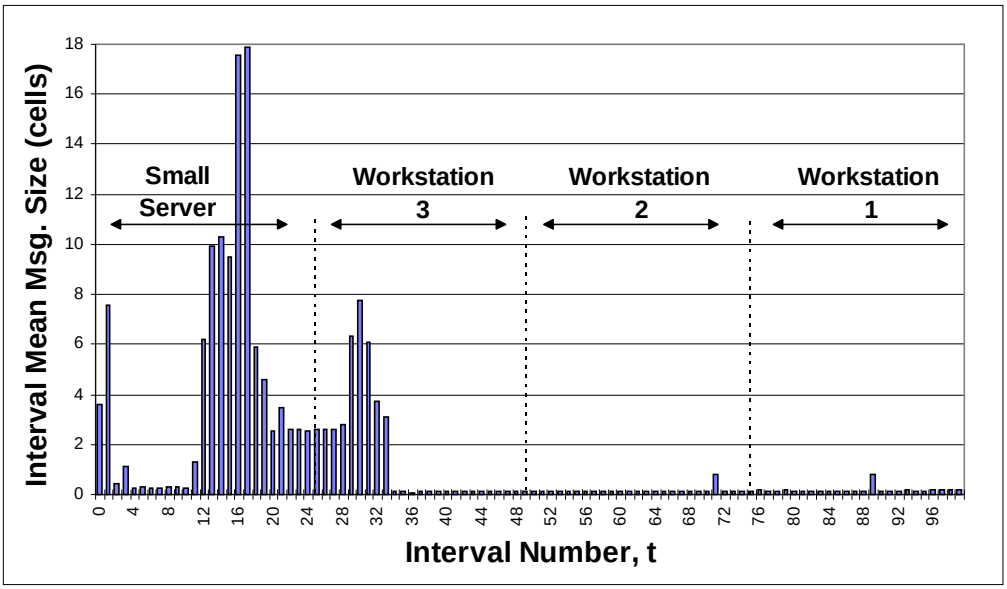


Figure 2.6: Composite "kp176.dat" Trace

Note that many of the individual readings within the trace are zero, representing 1 second time intervals when no outbound Ethernet packets were recorded. Obviously in such cases the trace reading algorithm of

our simulations would not generate a message. Hence, when calculating message size statistics, only the non-zero readings should be considered, as it is only these which generate actual messages. To this end, both the interval reading histogram (Figure 2.7) and the message size histogram (Figure 2.8, overleaf) have been generated. The former takes into account every recorded value, whether it be zero or greater, whereas the latter histogram only looks at the make-up of the non-zero readings (which are considered actual messages).

As will be explained later in Chapter 6, some particular simulation runs did require a composite trace which had a higher activity level throughout a greater portion of its length. This was achieved by producing the *"kp176half.dat"* trace, which was, exactly as the name suggests, the first 50% of original trace readings, in the same sequence (to retain the fractal properties). The histograms for both the full and half versions of the composite trace, along with mean and variance values, are depicted in Figure 2.7 and Figure 2.8 respectively.

Autocorrelation and self-similarity of the traces was also investigated. Namely, the issue of ordering and degree of spread of the traffic readings, and how these affect the ideal multiplexer and protocols was probed by also using as input the shuffled versions of the *"kp176.dat"* and *"kp176half.dat"* data traces. The shuffling process was a completely random redistribution of traffic readings to other positions within the file. However, the shuffling was not done on the entire file simultaneously - rather, each station's non-overlapping file portion was individually shuffled in order to retain the same overall load distribution, and destroy the autocorrelation of each individual station's traffic stream (from here referred to as the *intra-station* correlation). We have already seen from Sections 2.1 and 2.3 the significance of the need to know about traffic self-similarity characteristics. In light of teletraffic engineering's relatively recent "fractal revolution" which has spawned many works of research that confirm a high degree of self-similarity for LAN and data traffic in general, it is important to quantitatively describe the exact degree of self-similarity, for each of the traffic streams which are planned for use in the detailed testing of the F-CPR protocol in the latter part of this section.

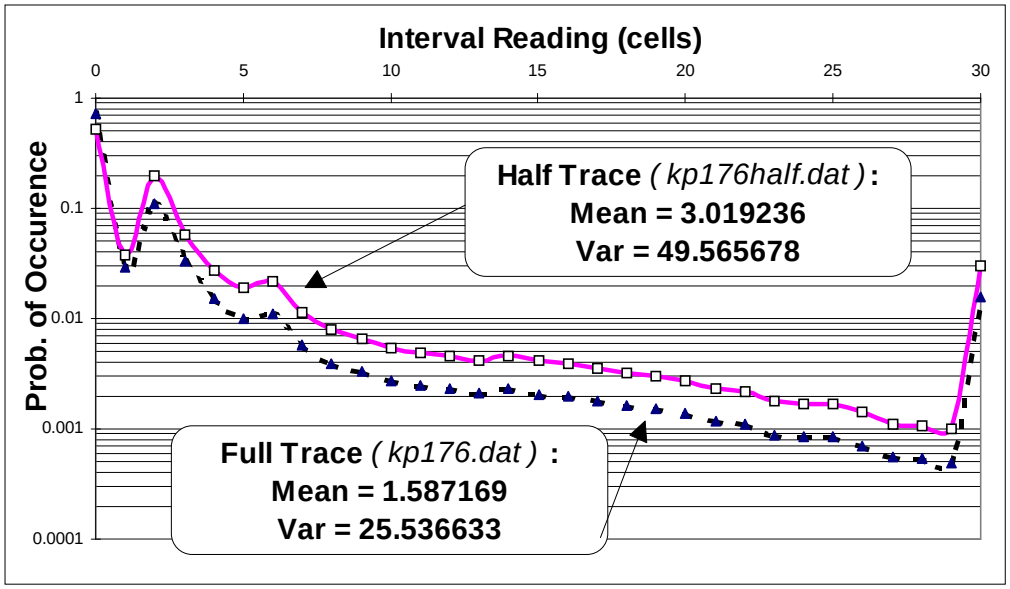


Figure 2.7: Trace Interval Reading Histogram and Mean/Var Stats

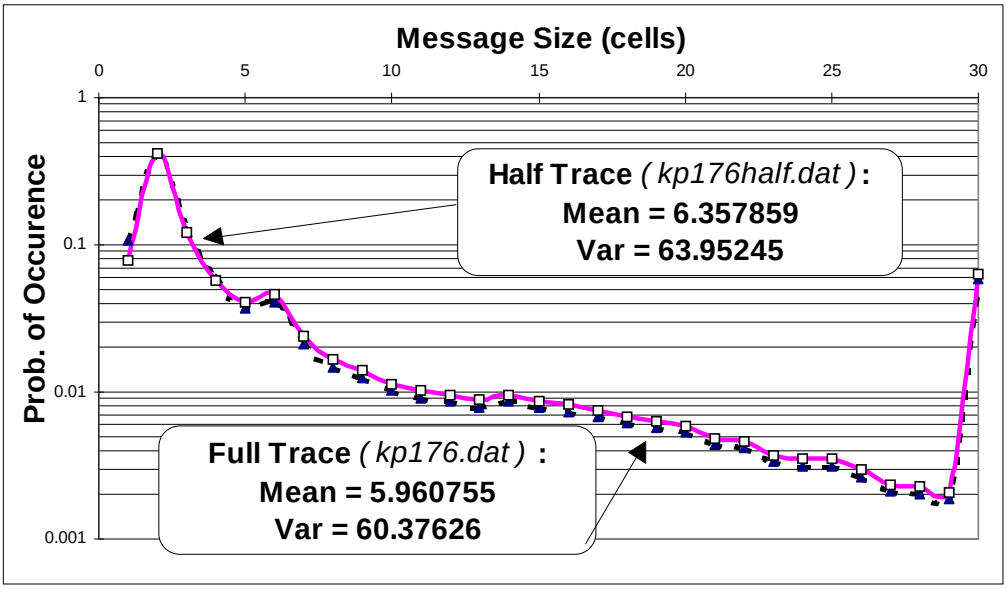


Figure 2.8: Message Size Histogram and Mean/Var Stats

To this end, using the technique described in [ADDI 95] we have plotted Interval Variance (of number of cell arrivals per interval) versus Interval Size in Figure 2.9, for each of the relevant traffic streams. The relationship between these quantities is given by the equation (2.9), which is just another form of equation (2.1) with $\alpha = \sigma^2$ (the unit-interval variance) and $\beta^* = 2 - \beta$.

$$Var(X(n)) = \alpha \cdot n^{\beta^*} \tag{2.9}$$

In order to obtain the Hurst parameter, the commonly used measure of self-similarity we defined in Section 2.1, it was necessary to perform a Log-Log transformation on the data points as shown in equation (2.10):

$$Log_{10}Var(X(n)) = Log_{10}\alpha + \beta^* \cdot Log_{10}n \tag{2.10}$$

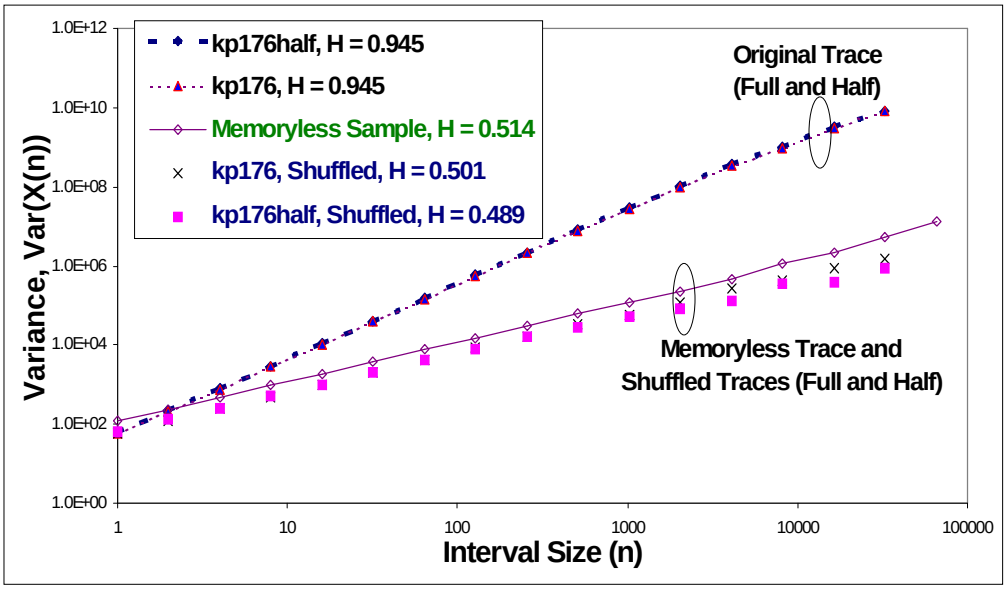


Figure 2.9: Hurst Parameter of Various Traffic Streams

Figure 2.9 is shown on Log-Log axes, in order to show that a linear regression is then done through the Log-transformed data points, with the aim of obtaining the gradient, β^*, of the resulting linear function. A graphical representation is important in this method of obtaining *H*, because it allows one to visualise which subset of points most closely resembles a perfectly straight line. The linear regression is then performed through these points. It is usually the few points at the upper edge of the line which are avoided, because they "taper off" in gradient due to the number of sampled intervals becoming very small (since the intervals themselves are so large) and thus an unrepresentative sample for calculations. Because the method of calculating *H* relies on graphical estimation, as well as least-squares linear regression, all of the *H* values we have determined cannot be assumed as exact. The Hurst parameter is given by equation (2.11):

$$H = \beta^* / 2 \tag{2.11}$$

We have just described the *Aggregated Variance Method* of obtaining *H*; this is probably the simplest and least accurate of a number of available alternatives, presented by Taqqu, Willinger and Teverovsky in [TAQQ 95]. There are three main classes of *H*-estimation techniques: block-aggregation, rescaled adjusted range (R/S), and periodogram. The R/S and Whittle's periodogram-based method are probably the most well known techniques, with the latter, *Whittle's estimator* technique being the only non-graphical method of obtaining *H*. It does however provide one of the most accurate *H* estimates and gives (very tight) error bounds - something which is very difficult to obtain with the Aggregated Variance Method we used here. However, our goal was to obtain an idea of the self-similarity properties of the trace without an emphasis on absolute accuracy in the estimate.

The information we obtain from the figure is arguably intuitive - the Memoryless Sample stream and the two randomly shuffled Real Traces were expected to show no or very little autocorrelation and self-similarity. In all three cases, $H \approx 0.5$, which according to theory suggests a traffic stream equivalent to an ideal, memoryless and non self-similar Poisson-distributed arrival process. The Memoryless Sample is derived from exactly such a process anyway, whereas the two shuffled Real Traces have had all the autocorrelation contained within them destroyed by the random reshuffling of their constituent samples. It is interesting that the *H* calculated for the shuffled version of the *"kp176half.dat"* trace is just slightly less than 0.5 (0.489); the reason for this observation could be either (a) a measurement underestimation error, or, (b) a very slightly negatively correlated sample resulting from the shuffling process. The close proximity of *H* to 0.5 in this instance (0.489) would indicate such a negligible negative correlation effect, that it is more likely that the 0.489 value was obtained through a measurement underestimation error, reason (a), rather than reason (b). However, it is the latter of these two reasons, (b), which is the more interesting one.

Namely, as explained in [LAVA 96], values of *H* smaller than 0.5 indicate negatively correlated traffic, examples of which may be generated in a variety of video coding equipment. In such a case, after each large message the probability of obtaining a small message is high. Similarly, the probability of seeing a large message after a small one is equally as high. Conversely, in positively correlated traffic it is the other way around, with small messages likely to follow small messages and the same applying for large

messages. With perfectly memoryless traffic, that has $H = 0.5$, such as that generated by the Memoryless Bernoulli-Geometric model, used in [SALA 96a] and later described in Section 6.1.2, there is absolutely no relationship or dependence between the size of one message and that of the next. Regardless of the previous message's size, the current message will always have a size conforming to some pre-defined distribution (e.g. 20% chance of a 30 cell message and 80% chance of a 1 cell message).

Although it was expected that the original trace, in both its full and halved forms, would exhibit a substantial degree of self-similarity, it is noteworthy that the Hurst parameter is as high as 0.945 - a figure which almost represents the worst possible degree of self-similarity (remembering that $H = 1.0$ is the maximum). This is something one obviously cannot control when collecting real traffic Traces, so it is a fortuitous outcome because it means that we will be ***able to test the F-CPR protocol with two extreme types traffic, self-similarity wise***. It also means that the real traces which are used to load the protocol will have almost the worst possible self-similarity artefacts, and will hence show the most conservative delay and utilisation performance. Testing systems with worst-case traffic arguably yields the most satisfactory (read conservative) dimensioning guidelines, from a true engineer's point of view.

3. Selected Modern High Speed Data Protocols

3.1 Fairness and Congestion Control in Data Protocols

Zukerman and Chan [ZUKE 94] argue that the responsibility for achieving global fairness (a user getting no more than a fair share during overload) as well as controlled congestion is the responsibility of the network end-user. In their scheme, it is highlighted that this can be achieved without needing a central controller, or each node having exhaustive traffic flow information from other nodes.

3.1.1 Congestion - Definitions

Network congestion causes the overflow of buffers and hence loss of packets. When data is lost, retransmissions will be required in order to provide data integrity. However, the throughput is effectively reduced when retransmissions take place. In some end-to-end protocols, the effective throughput reduction is more severe, because the protocols retransmit in batches or blocks, both the lost and successfully transmitted packets.

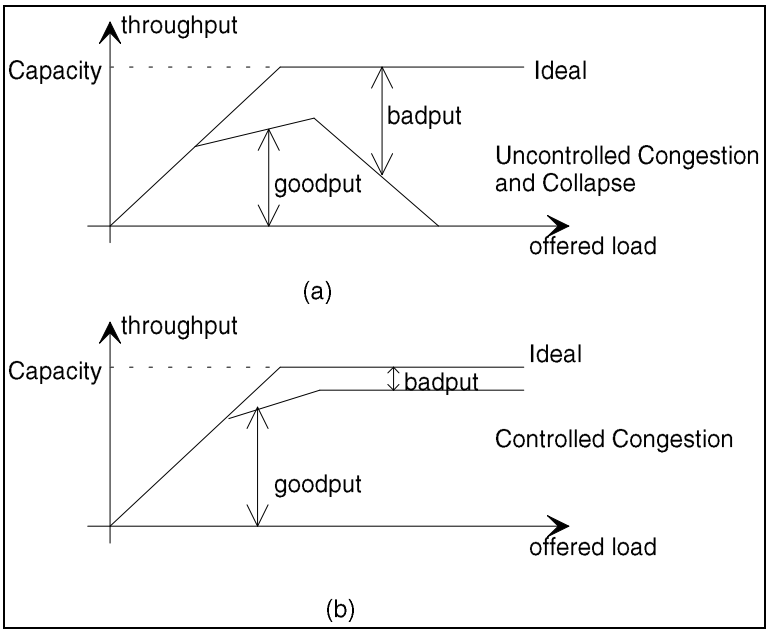


Figure 3.1: (a) Network behaviour under uncontrolled congestion.
(b) Network behaviour under controlled congestion.

With reference to Figure 3.1 and using the definitions in [ZUKE 94] and references therein, we can define *goodput* as the useful throughput of a network. As illustrated in Figure 3.1(a), under increasingly heavier loads, and without the concept of congestion control, this useful throughput keeps decreasing until finally all bandwidth is consumed for retransmissions - a phenomenon known as *congestion collapse*. The role of *congestion control* is therefore to reduce the amount of useless retransmissions (defined in the figure as

badput) and maintain the goodput at the reasonable level shown in Figure 3.1(b). Congestion control may be achieved in one of two ways:

- Users only retransmit lost packets - *selective repeat* protocols. Considered somewhat complicated to implement and most common transport protocols (e.g. TCP, OSI Transport Protocol Class 4) do not use it.

- Users implement some *adaptive control schemes* which reduce the load when congestion is experienced.

With reference to the latter point, currently supported versions of such schemes can achieve congestion control only if ALL users act in a co-operative manner. Namely, it is still possible for a malicious user to continuously send packets, causing the other compliant users' adaptive control algorithms to detect a high network utilisation and thus back off. [ZUKE 94] claims that it is therefore essential for the network operator to provide incentives to users implementing adaptive load control. That is, to introduce mechanisms which would ensure that under overload conditions, no user gets more than a "fair" share of network resources, as detailed in Section 3.1.2. If some users still fail to adapt their rates when network congestion occurs, then they will be the only ones to suffer without affecting the goodput of the cooperative users [O'NEI 92].

3.1.2 The Fairness Criterion

The [ZUKE 94] notion of fairness refers to each end-user being treated according to the maxmin [BERT 92] criterion (with additions described below), which basically attempts to maximise the throughput of end-users with the minimum bandwidth requirements, while restricting the throughput of the end-users with the highest capacity requirements. In this way, no user achieves a higher throughput than a controlled user.

The maxmin-based fairness criterion described in [ZUKE 94] has the following features, which achieve controlled congestion as well as global fairness:

1. Allows for uneven (prioritised) apportionment of capacity,
2. Penalises only those users who cause overload,
3. Relates to throughput fairness, rather than delay fairness, and
4. Achieves good efficiency, since all capacity is used.

This method of achieving fairness is somewhat intuitive, because if the high-capacity requirement sources (another term for end-users) were allowed to transmit with not-so-stringent throughput controls, there would be no capacity left within the network in a matter of just a few connections. The examples of Medium Access Control (MAC) Protocols for Hybrid Fibre/Coax (HFC) networks which are given in Section 3.3.5, also implement a similar type of maxmin fairness.

Of the four features belonging to the described fairness criterion, the most significant is the first one. It is important to be able to allow for non-equal capacity apportionment, because in practice, some sources such as file servers or multiple-connection terminals have a need to use more bandwidth than other sources. Under overload, it may be desirable to allocate more bandwidth to such sources. In order to be able to differentiate sources by relative bandwidth requirements, [ZUKE 94] introduces a number called the *relative usage value* (RUV), which may be allocated to each source permanently, or at each call set-up and take-down instant.

Using the sources' RUVs and offered loads, it is possible to define two distinct groups of sources: (1) uncontrolled, which receive a bandwidth allocation equal to their offered traffic, and (2) controlled, with lower throughput than their offered traffic. The main rule of this fairness criterion is that,

- *The relative throughput (source throughput divided by RUV) of a controlled source will not be lower than that of an uncontrolled source sharing its bottleneck link.*

For the case of a general network, one may have the impression that achieving fair apportionment requires a central controller. [ZUKE 94] shows that this is not the case, and that the fairness criterion may be achieved by combining the *fairness discarding algorithm* of [O'NEI 92] implemented at each link, and an adaptive flow control scheme like the one proposed by Jain in [RAMA 88]. In this way, the overall scheme to achieve general network fairness works in three steps:

- Firstly, the already described methods are used to determine controlled and uncontrolled sources' relative throughputs, and thus to initially share the capacity of each link.

- Secondly, on a link basis, the *fairness discarding algorithm* would begin discarding incoming cells from a particular source deemed to be exceeding its fair apportionment of that link's capacity.

- Lastly, the adaptive flow control scheme, would enable each source to customise its offered traffic to the network, so that it converges to that allowed by the bottleneck link of an individual source.

If each end-user implemented the above set of rules, then any instances of congestion would be controlled (evenly spread quality of service degradation among all sources), and fairness would be maintained **both locally and globally**. Of course, the problem of congestion collapse would be confined to any user not using this fairness algorithm to adapt their load to congestion situations. This is desirable, because rather than degrading everyone else's quality of service, it is only the end-user who violates the fairness set of rules, that suffers congestion collapse.

3.2 Access Network Architectures and Multiaccess Techniques

There are two diametrically opposite multiaccess paradigms used for sharing a communication resource among many competing users: (i) the *reservation-based* multiaccess approach, which may be compared to circuit switching; and, (ii) the *random multiaccess* approach, which is akin to the philosophy of packet switching.

Reservation-based techniques rely on the exclusive use of a portion of the communication resource, by only one user. For this reason, they tend to subdivide the communication resource at the physical layer, into frequency separated channels, or different timeslots within some framing structure. This method, while providing absolute guarantees of bandwidth availability on demand, tends to be inefficient especially if the user only sends information intermittently. The three main reservation-based techniques available to today's access network designers are Frequency Division Multiplexing (FDM), Time Division Multiplexing (TDM), and Wavelength Division Multiplexing, (WDM) although various combinations of these are often used to further segment bandwidth into “chunks”. The usability of any of these techniques depends largely on the physical layer medium of the access network in question. For example, WDM is not feasible on the coaxial cable part of an HFC network, because the signal is obviously no longer in lightwave form.

Taking a completely different philosophy in sharing the available bandwidth, random access, or packet-switched, techniques rely on an unsegmented, shared physical transmission medium and the application of some added access control mechanisms (usually at a higher layer). There are two sub-categories of such techniques: (a) algorithms which avoid collisions, via polling or demand based scheduling; and (b) algorithms which cope with collisions using, for example, a Slotted Aloha type contention resolution algorithm. Either approach achieves the original goal of transmitting the information of many users on the single shared physical medium. It is significant that collision-avoidance variants of the random access paradigm (sub-category (a)) are in fact emulating the structured philosophy of the reservation-based multiaccess approach, but are doing so on a physically shared medium (e.g. on the same frequency channel or timeslot). One such technique belonging to sub-category (a) is *demand-assigned multiple access (DAMA)*, where the underlying method of access is based on packet switching, but the scheme reacts to the users’ individual demands and then reserves bandwidth for each station fairly and in a way so as to avoid collisions. A similar but less complex sub-category (a) member is *polling*, where bandwidth is statically and cyclically allocated to users regardless of their demand (early host-terminal networks were based on this type of polling).

Regardless of the variations employed, contrary to circuit-switching, bandwidth guarantees for either sub-category of the packet-switching approach cannot be absolute (at best, they may be within some tolerance bounds if a priority scheme is used); however, use of the packet-switching approach tends to utilise the shared medium more efficiently system-wide, leading to less bandwidth wastage.

There are many systems where these two multiaccess philosophies are used in conjunction with one another. One simple example is an ISDN B channel (64 kbit/s) being used to carry multiple stations’ Internet Protocol (IP) data packets from, say, a branch office back to headquarters. In such a scenario, the ISDN channel is dedicated solely for the customer, who happens to have multiple stations on-site. The simultaneous carriage of multiple workstations’ IP packets represents a dynamic subdivision of the original dedicated channel. Although the interaction between the reservation-based and random access paradigms happens, in this simple example, between different layers of the protocol stack (i.e. at the data link layer and the network layer), the example is nonetheless a case of packet-switching overlaid on top of, and used together with, circuit-switching.

Like in this narrowband example we have given, many proposed designs for broadband access networks also rely on combining the circuit- and packet-switching paradigms. Access network architectures using both TDM and random multiaccess, offer a good trade-off between the savings on cheaper optical components and the added complexity of a MAC protocol, according to [PITT 93]. High-density WDM solutions are expected to compete seriously with TDM in the near future, in terms of available bandwidth, as technological advances are made [OAKL 91]. The principle of combining the reservation and contention based modes of sharing the medium is nonetheless expected to remain popular (but not a given in all systems) regardless of the particular component schemes used.

We now look at some of the currently most popular, and in some cases competing, access network architectures. It is important to note that some of these access network architectures can employ either multiaccess paradigm, or both. A good example is Fibre To The Home, which may be based on dedicated paths to each residence or some type of optical bus system.

3.2.1 Architecture and Topology

An important issue in the design of customer access networks (CANs) has been the depth of the optical fibre's penetration. Namely, should we build Fibre To The Home (FTTH), Fibre To The Curb (FTTC), ADSL-based or HFC layouts. As explained in [SLOS 96], in early stages of their use optical fibres were used as high bandwidth inter trunk-exchange transmission medium. In today's world of increasing demand for broadband communications and distributive services (i.e. the infamous "Information Superhighway" in the form of home Internet access or interactive Cable TV) the use of high-bandwidth fibre technology even as far as the customer premises, has long been considered an option. To complete the ensuing discussion on choice of access network architecture, we also consider the option of providing access to the customer with no fixed, wireline infrastructure; that is, of using an access topology based on wireless radio.

3.2.1.1 FTTH Architecture

The construction of a CAN using an FTTH architecture is the most expensive option for network operators and one which we will eventually come to, but not in the immediate future. It is still unlikely that today's customers require a *dedicated information pipe* hundreds of megabits per second "wide", although this may very well be the trend of the future. As its name implies, FTTH brings the optical fibre right to the customer premises equipment (CPE), and a number of access network topologies are possible, as illustrated in Figure 3.2 overleaf.

In all CAN configurations except for (c), a single medium is used for both directions of transmission. In such systems, WDM-capable equipment would need to be used with two wavelengths - one for each direction of transmission. System (a) shows an active star - the simplest topology, since every link is dedicated to one subscriber, and requires one optical receiver and transmitter at both the local exchange (central office) and at the customer premises. On the other hand, the active single bus in (b) requires only

one pair of optical components at the local exchange. Two passive slotted contra-directional buses, similar to those used in Distributed Queue Dual Bus Systems (DQDB) systems, are depicted in (c). System (d) is a passive single tree, using optical splitters at the demultiplexing points and once more only one receiver-transmitter pair is required at the exchange. Finally, (e) is a topology where a single active ring, requiring a single wavelength, passes through all customer premises, and requires just a single sense of transmission (i.e. "ring"). The term *active* is used for actively powered optical receivers and transmitters, while *passive* refers to a passive frame write mechanism (see [SLOS 96] for details).

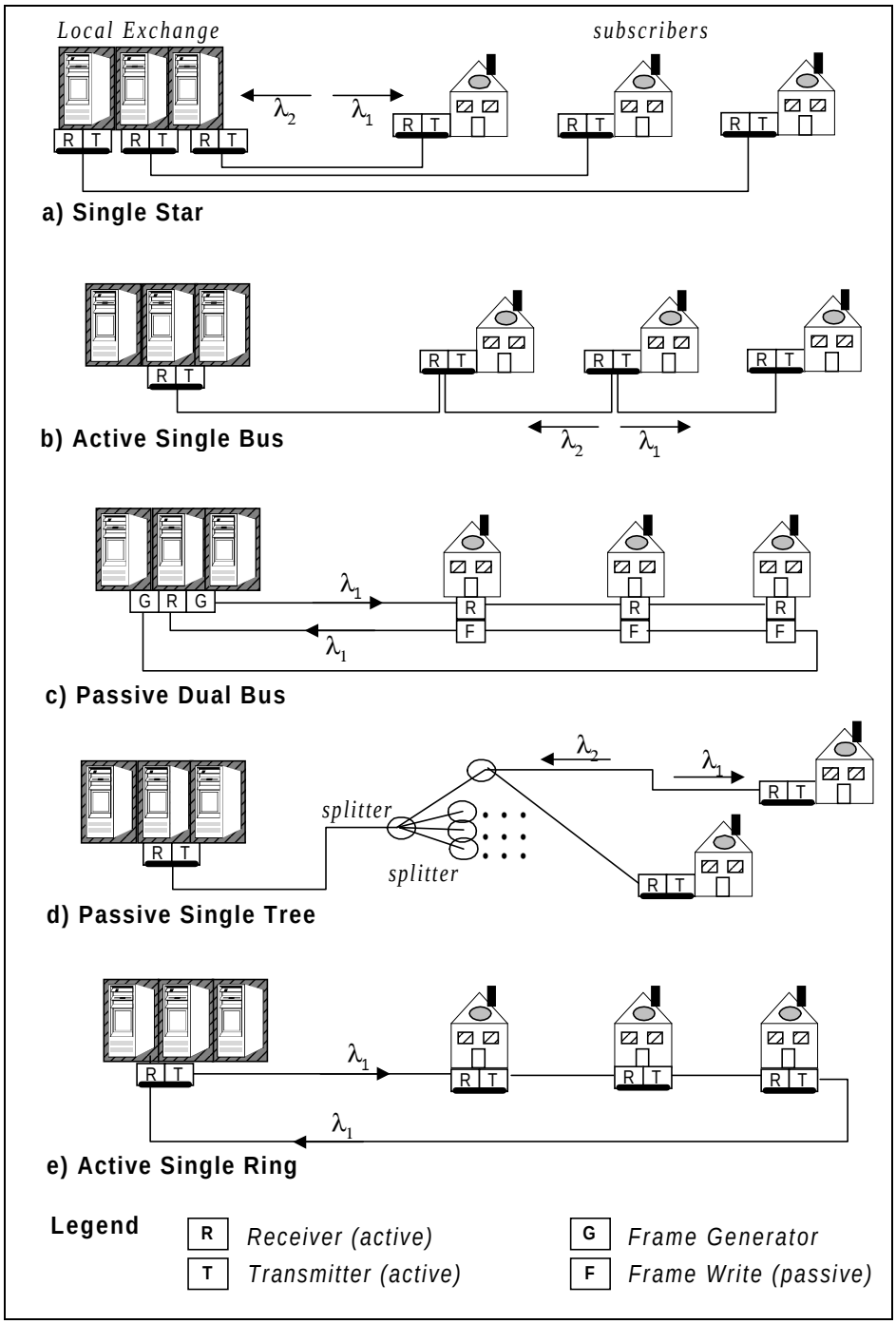


Figure 3.2: Active and Passive FTTH Topologies

3.2.1.2 FTTC Architecture

A few telecommunications companies (telcos) have decided to push the FTTC architecture, and implement CAN upgrades so as to provide dedicated point-to-point connections to all subscribers, to re-use existing twisted pair subscriber lines, or to provide Personal Communications Systems (PCS) from fibre nodes, according to Monti [MONT 95]. The paper by Monti also states that the chief difference between FTTC and HFC architectures is a "generally accepted" one: that FTTC requires a higher cost to provision. It is mainly for this reason that most cable and telecommunications companies have opted to invest in HFC CANs (see Section 3.2.1.4 below).

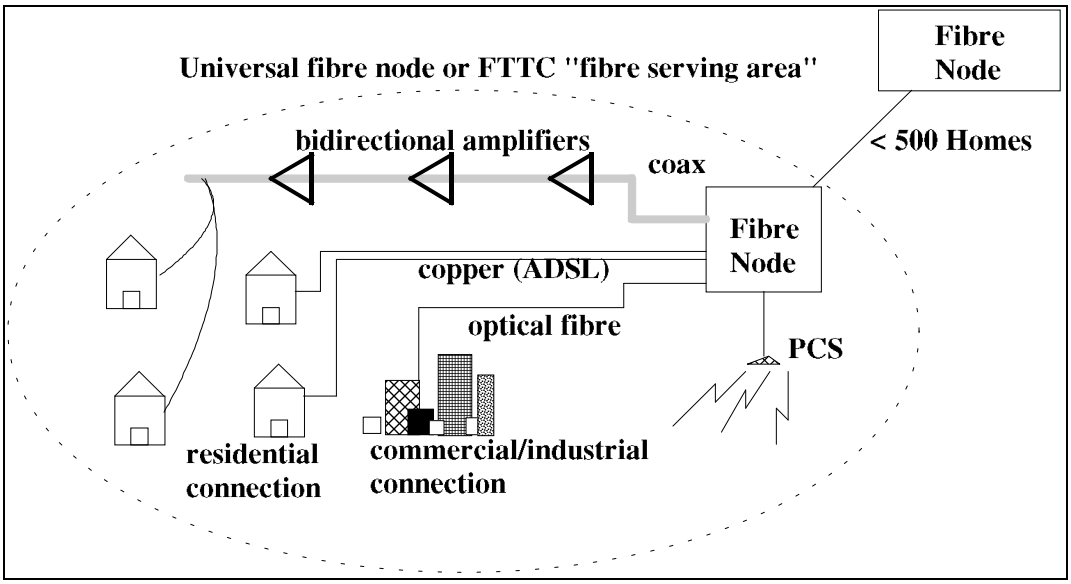


Figure 3.3: FTTC Access Network Segment

Figure 3.3 shows a typical FTTC access network segment. [MONT 95] points out that a number of technologies over various media may be used for the "subscriber drop", a term coined for the last few hundred metres from each curb fibre node to the customer premises. As is evident from Figure 3.3, copper twisted pair (using Asymmetrical Digital Subscriber Line technology - ADSL) wire, coaxial cable, wireless PCS access and optical fibre can all be used for the subscriber drop. Note that it is envisioned that for this architecture, the technology gaining most widespread acceptance is ADSL and its variants, such as HDSL (High-data-rate DSL, symmetrical at 1.5Mb/s) - until the day that fibre is laid out all the way to the home.

The other key differences of FTTC relative to HFC architectures are smaller node sizes (24-500 households passed per serving area), baseband digital modulation of the fibre, digital switching and dedicated port electronics in the fibre node (i.e. one pair per subscriber), dedicated point-to-point use of media downstream of the fibre node (dedicated channel for every subscriber means that after the node, each subscriber must have own separate transmission path) and higher total plant mileage. Taking all of these into consideration, it only follows logically that it is a generally accepted fact that FTTC is a topology which has a higher provisioning cost.

3.2.1.3 Asynchronous Digital Subscriber Line (ADSL) -based Architectures

ADSL-based architectures can be thought of as the precursor of the telcos' push to get fibre optic transmission as deep into the CAN as FTTH or FTTC architectures. The latter two involve the penetration of fibre as deep as the customer's premises, or, at least closer to a customer than their local telephone exchange. On the other hand, first generation ADSL systems are envisaged to fully utilise the existing twisted pair copper plant running all the way from the local exchange out to the customer's residence (a distance of up to 6.6 km). This has the implication that fibre need only go out to all local telephone exchanges, something which is already a reality today (for almost all major telcos in the developed world).

When Bellcore first conceived the ADSL in 1989, it was defined with downstream rates of 1.5Mbit/s over 6km of 24-gauge wire and 6Mbit/s over 4km of 24-gauge wire [MAXW 96], with upstream rates up to 640kbit/s depending on line quality and length. In early 1996, it was observed that ADSL had a greater chance of success if it tried to satisfy demand for Internet access rather than for video to the home. As the Internet is inherently variable rate, it was suggested to make ADSL also a variable bit-rate technology, with interfaces to the network or home PC to be either ATM or Ethernet (variable rate anyway). In this way, the subscriber could be offered the best rate their line would allow at the given time, without going to the extremes of adapting to the smallest of variations (hence the term *rate adaptive* was preferred over *variable rate*). Namely, even if the downstream rate fell to below 1.5Mbit/s, the user would still have a service with delivery speed far in excess of what the Internet backbone would be running at anyway, for the next few years. The logic of this idea, coupled with the beneficial property of telcos being able to increase the number of ADSL-capable lines without modifications, has ensured the remarkably rapid endorsement of rate-adaptive DSL (RADSL) by telcos. It is foreseen in [MAXW 96] that in the near term, the United States market will be dominated by RADSL modems, while fixed rate ADSL may remain sufficient for countries exclusively targeting the technology for video delivery.

Currently, there are two types of competing ADSL modulation techniques making their way to the marketplace, and these are **(i) single-carrier** and (ii) **multicarrier modulation**. Each of these types has one or more specific candidate techniques under it, and these are discussed in Section 3.2.1.3.2. Note that a system running either of these techniques, may be operated in one of two modes of ADSL operation. Figure 3.4 (overleaf) shows the **two main modes** of ADSL operation: Frequency Division Multiplexing (FDM) and Echo Cancellation (EC). Note that both modes filter out the lowest 25kHz of the spectrum, even though the POTS needs only 4kHz. This is due to the edge filters (also termed *POTS splitters*) which become exceedingly difficult and expensive to design as the lower upstream spectral edge approaches the POTS band.

3.2.1.3.1 Upstream/Downstream Channels and Transmission Rates

The upstream channel is a part of the spectrum with about 135kHz of usable bandwidth, as shown in Figure 3.4. This part of the spectrum has the most favourable channel attenuation characteristics, but suffers the most crosstalk from other services (e.g. ISDN interferes up to 80kHz, high-bit-rate DSL (HDSL) interferes up to 240kHz).

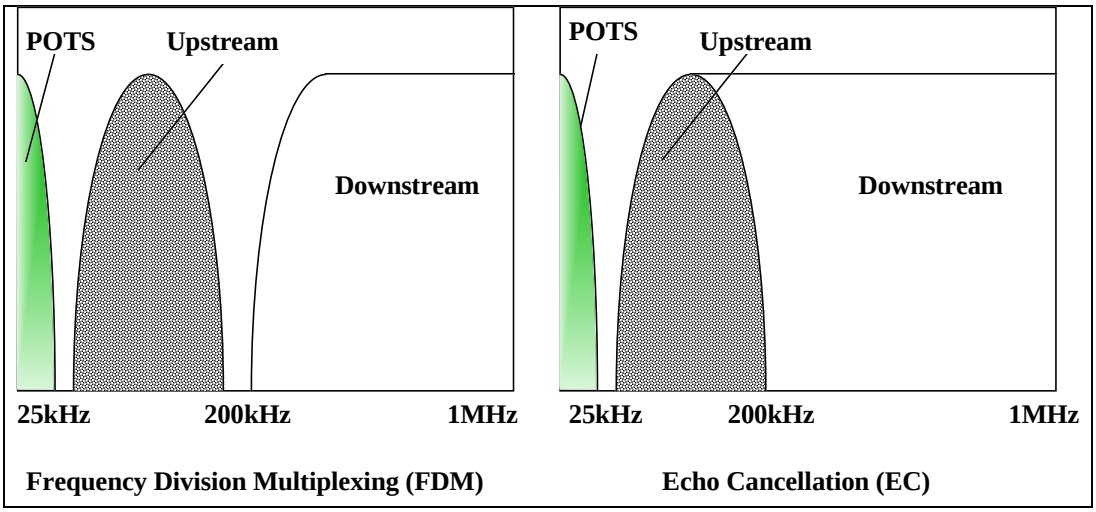


Figure 3.4: Modes of ADSL operation (channel configurations)

The left portion of Figure 3.4 illustrates the channel configuration for the frequency division multiplexing (FDM) mode of operation. The downstream channel starts above the upstream at about 240kHz, and then extends as far up as needed, or permitted, by a combination of the desired data rate, attenuation, and modulation method. On the other hand, as is evident from the right side of Figure 3.4, in the echo-cancellation (EC) mode, the downstream channel overlaps the upstream. For the added cost of the echo canceller, this provides two advantages over the FDM mode: (i) the downstream has access (at certain times) to the higher quality part of the spectrum, and (ii) the upstream band may be extended upwards in frequency without being limited by the edge of the downstream band. So far, only multicarrier ADSL modems have been implemented with the EC mode. The latest proposed *rate-adaptive single-carrier system* will use downstream channel frequencies from 631kHz to 1.491MHz, with adaptive downstream bit-rates ranging from 640kbit/s to 8192kbit/s and upstream bit-rates ranging from 272kbit/s to 1088kbit/s. Granularity: only 5 coarse steps from highest to lowest rates [MAXW 96].

A *rate-adaptive multicarrier ADSL system* uses a band from 240kHz to 1.1MHz for the downstream channel, with the downstream bit-rates ranging from 32kbit/s out to 9Mbit/s, in increments potentially as fine as 32kbit/sec. The upstream channel bit-rates vary from 32kbit/s to 1Mbit/s. Granularity: very fine rate increments of 32kbit/s for both channels.

3.2.1.3.2 ADSL Functional Layers

Initially ADSL was conceived to be a relatively unintelligent bit-pump with a 1.5Mbit/s downstream rate for video delivery and a small 16kbit/s duplex channel for signalling and video controls [MAXW 96]. This simple bit-pump concept has since been dropped in favour of a suite of features, considered by relevant ADSL groups *standardisation along the lines of typical ATM physical (PHY) layer protocols, such as [ATF1 96]*. Namely, the basic modem functionality is to be divided into two sublayers - the physical-media-dependent (PMD) sublayer and the transmission convergence (TC) sublayer. In Figure 3.5 we see the functional divisions for each of the sublayers, and underneath these, are listed the ***layer's implementation options.***

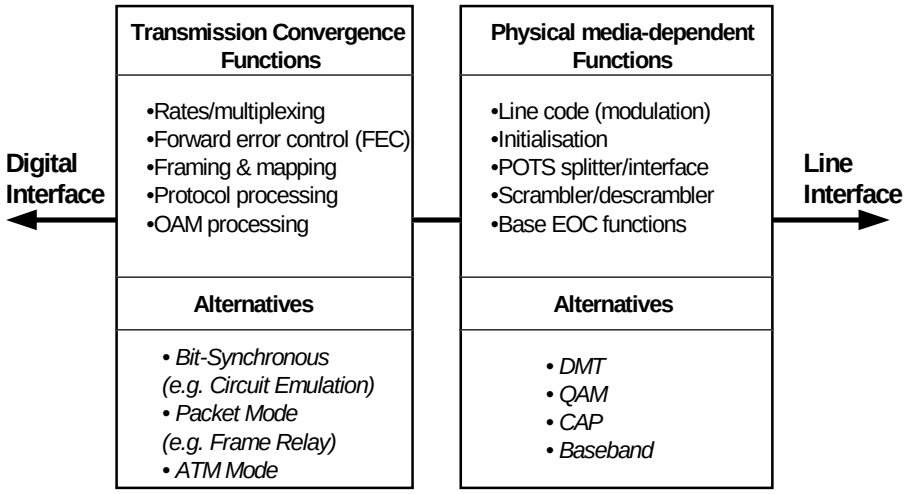


Figure 3.5: ADSL functional layers

Options in the ADSL Physical-Media-Dependent Sublayer

At present, three modulation techniques (line codes) are making their way to the marketplace. Here we list a brief summary of each.

(1) Discrete Multitone (DMT) is a multicarrier technique, dividing the line into many small channel portions, and modulating each one separately, based on its capacity. ANSI standards group T1E1.4 has developed a standard, number T1.413, based on DMT [ANSI 95].

(2) Quadrature Amplitude Modulation (QAM) is the "parent line code for all ADSL". QAM has the best combination of bandwidth efficiency, performance in the presence of noise, and timing robustness, than any other scheme (e.g. 2B1Q used in ISDN and HDSL). Because of the widely ranging qualities and characteristics of telephone lines, for ADSL it is the adaptive equaliser which dominates the system complexity and cost for QAM implementations.

(3) Carrierless AM-PM (CAP), is a variation of QAM developed by AT&T, which generates a transmit waveform by applying each half-rate bitstream to a pair of digital transversal bandpass filters with equal amplitudes but phase responses differing by $\pi/2$. This produces the same spectral shape as QAM and has almost identical performance as QAM, but has some added efficiencies with digital implementation, when compared to QAM.

According to [MAXW 96], "the market for ADSL will witness over the next year or longer, an ongoing ... fight between DMT and CAP" modulation techniques. It is therefore instructive to look at the advantages one has over the other. The T1.413 standard, adopted by the ADSL standards group, embraced DMT over CAP in 1993, because of the following advantages DMT enjoys over the single carrier CAP technique:

- Optimises performance over a wider range of lines and data rates than CAP.
- Operates over more lines at a given rate than CAP.
- Runs at a higher data rate over a given line than CAP.
- Provides higher immunity to impulse noise than CAP.

In its favour, CAP has the advantages that:

- It is a more mature, better understood technology.
- It is presently available in highly integrated form with lower power consumption and price.
- Enjoys the majority of field trial installations.

Options in the ADSL Transmission Convergence Sublayer

A very important factor in the development and market penetration of ADSL is how well and how soon it can match the varied and changing protocol environment it must meet. The following interfaces are either already provided by ADSL modems or will be in the near future:

(1) Bit-Synchronous - the T1.413 ADSL standard maps a set of serial interfaces, built on the North American Digital Hierarchy, arranged to fit various line lengths. High-speed simplex downstream channels (1.536Mbit/s - 6.144Mbit/s) as well as low-speed symmetric bidirectional channels (64kbit/s - 640kbit/s) are part of the specification. The physical interfaces include T1, V.35 and RS232.

(2) ATM - this interface would involve the ADSL modem performing ATM TC layer functions (e.g. cell delineation, rate decoupling, header error correction (HEC) generation and verification, and framing. The modem rate will be adaptive, with DMT having rate granularity of 32kbit/s and CAP having 320kbit/s. The physical interface will most likely be ATM Forum 25Mbit/s, making the modem, which operates at a slower rate, the point of congestion.

(3) ATM/Ethernet - initially, this is the most likely interface, with telcos opting to deploy access networks with ATM to the premises ADSL modem, where the service would encounter a non-ATM interface PC. Three possible configurations may solve this problem: (i) ADSL modem installed inside the PC, eliminating the need for an external interface. (ii) ADSL modem will tunnel ATM cells through an Ethernet interface, using some protocol like Cells In Frame (CIF) (see [ARM 90], [ARM 93] or [CIF 96] for more details on CIF). (iii) ADSL modem will connect via the new universal serial bus interface, available on all Intel-compatible computers from 1997.

3.2.1.3.3 Multplexing of ADSL Lines at the Local Exchange

From the diversity of solutions available in the marketplace [ADSF 97], it would appear that the traffic multiplexing method at the local exchange is going to be dependent on (i) the vendor, and also on (ii) what protocol the vendor's ADSL equipment is designed to support at the higher layers. For example, different vendors are already pushing two distinct types of traffic aggregation methods at the local exchange:

1. The first relies on a system supporting a bridged/routed protocol over ADSL, and it terminates with an **Ethernet style bus interface** at the local exchange, into which the user ADSL lines are connected.

2. The second type of solution involves an ATM-over-ADSL system, where an **ATM Access Device (AAD) acts as an ATM multiplexer unit** with or without switching capability, and sits in the local exchange.

3.2.1.4 HFC Architecture

3.2.1.4.1 Fast Enough yet Economical

Primarily for the cost reasons described in Section 3.2.1.2, many telecommunications and cable companies around the world have committed to HFC. As is evidenced by the strategies of both major carriers in Australia (Telstra and Optus), telecommunications companies are often paying for "complete overbuild/rebuild of subscriber drops for the locales in which they haven't installed coaxial plant to work from as a base" [MONT 95]. Figure 3.6 illustrates an HFC access network segment.

It is expected that the next cycle of the "cyber-revolution" will be the speedy uptake of interactive residential broadband/multimedia services. [BISD 96a] heralds the fact that Video-on-demand (almost a virtual stroll through the local video library without leaving your living room), full tele-commuting (utilising the entire office workstation's interface power remotely from home), remote learning, tele-shopping, World Wide Web surfing, and powerful interactive games are ready to enter people's homes, today, but not before "the 28.8kbps barrier of today's fastest modems is topped ... by 100-fold or even more, by a new breed of telecommunications technologies".

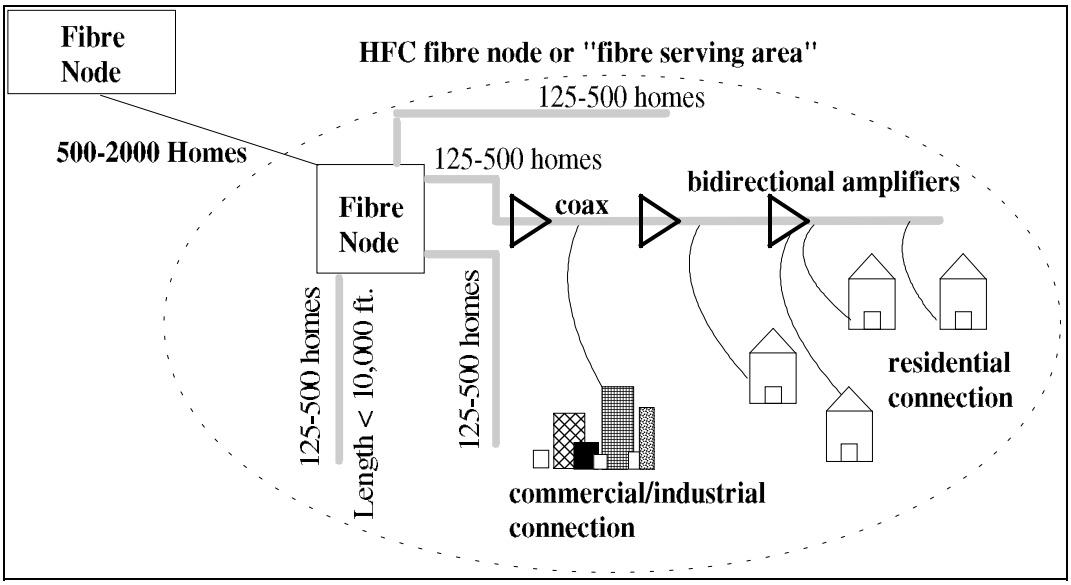


Figure 3.6: HFC Access Network Segment

As [SALA 96a] points out, this need for increasingly faster "information highway" on and off ramps has been met to date by the services currently available across the standard twisted pair physical medium, such as: data modems across PSTN lines, ISDN connections and more recently Asynchronous Digital Subscriber Loop technology [WARI 91]. However, it is the alternative medium of combined optical fibre

and coaxial cable, referred to as HFC which has emerged as an ideal candidate to appease the current demand for increasing bandwidth.

The reason for this suitability lies in the fact that today's existing cable TV (CATV) network infrastructure provides a very large excess bandwidth - well over 300 MHz, while the theoretical frequency bandwidth of the cable is in the order of a GHz, providing plenty of room for future generations of opto-electronic devices to improve the medium's carriage capabilities. In addition, the existing CATV networks can be modified relatively easily and cost-effectively, to provide the necessary two-way communication path required.

3.2.1.4.2 HFC - Dedicated or Shared Medium?

In the article by Whittle [WHIT 96] about Optus Vision's Hybrid Fibre/Coax telephony, Internet and video infrastructure, examples of both circuit- and packet-switching approaches for the HFC variant of access networks are given. Both of Australia's two major carriers, Optus and Telstra, began rolling out HFC systems in 1994, with Optus introducing Australia's first HFC telephony service midway through 1996.

The HFC network rolled out by the carriers is a good example of an architecture able to support both circuit- and packet-switched methods of medium sharing. Namely, the 5 - 65 MHz upstream and 450 - 750 MHz downstream frequency ranges can either be:

- Subdivided into *X* MHz channels (*X* = 1-2 MHz in the upstream, *X* = 6 MHz in the downstream) that then provide voice and data communications in single or multiple units of 64Kbps (the 'DS0' data rate which is the international standard for carrying telephone signals [WHIT 96]). **This is reservation-based medium access and is often referred to as "multicom" [WHIT 96].** In most manufacturers' systems (Optus Vision is using U.S. based corporations, ADC and Motorola as primary suppliers) each DS0 channel can be assigned to a particular customer in 50ms or so, and upstream and downstream channels can be assigned independently. The DS0 units may carry information from voice, facsimile or modem data communication. For example, Australian modem manufacturer NetComm has created the “NetRocket 512” interface, which is essentially a router and cable interface that can scaleably use between one and eight DS0 units to provide symmetrical data access rates of between 64kbit/s and 512kbit/s (dedicated).

  OR

- Subdivided into *X* MHz channels (*X* = 1-2 MHz in the upstream, *X* = 6 MHz in the downstream) each of which can be accessed by all users in a Fibre Serving Area (FSA) by means of combining TDM together with a regulated contention scheme. **This is hybrid scheme of medium sharing,** requiring the implementation of both frequency and time division multiplexing at the physical layers, together with a MAC protocol at the data link (and physical) layer.

Some interesting research has already been carried out on multicom systems, with Houck and Lai [HOUC 95, HOUC 96] looking at several key traffic issues such as (i) call packing into DS0 timeslots to improve upstream bandwidth efficiency, (ii) impact of frequency hopping proximity restriction on call blocking, (iii) optimisation of DS0 timeslot assignment algorithms and (iv) downstream load balancing. The multicom studies largely rely on a foundation of well-known traffic engineering concepts and formulas such as the use of the Engset formula with minor modifications, or state space analysis for efficient timeslot assignment algorithms. Conversely, teletraffic studies of shared medium HFC networks necessarily require the design and analysis of a complex protocol (and its plethora of add-ons and components) specifically suited to the salient physical features of such networks. Namely, a fully shared, contention-based medium requires a protocol with an efficient framing structure, method of control (centralised or distributed), bandwidth reservation capability, contention resolution algorithm, priority scheme, and scheduling scheme.

The suite of features just mentioned form the basis of a Medium Access Control (MAC) protocol tailored to HFC networks, which is one of the main focal points of our work. Most publications in the literature refer to shared medium systems as *HFC cable data modem (CDM)* systems. However, the term “data” in this context may potentially be a little ambiguous.

First generation HFC networks will use completely separate ranges of the frequency spectrum for telephony, video distribution, and cable data modem systems, and hence it may be assumed that initially the cable modem’s primary use will be as a fast access link to the Internet [WHIT 96]. But as both HFC and ATM technologies mature, and telcos start realising their full-service network dream, it is envisaged that cable data modems will evolve together with ordinary pay TV set-top units and household telephone connections into a universal home access unit for the HFC network. This unit would act as a point where various types of services could be aggregated onto a common ATM carriage platform for example.

The unit would thus be the equivalent of today’s cable modem and could employ much of the functionalities of the MAC protocols being proposed and standardised today, such as *prioritisation* (e-mail data versus a non-real time variable bit rate (VBR) service such as Internet video telephony) and scaleable synchronous *circuit emulation* (64Kbit/s circuit emulation for a telephony call or a 2Mbit/s MPEG digital video stream).

Turning our attention to the typical spectral allocation in HFC systems, we find that [BISD 96a] states that the downstream (i.e. from the CATV headend station or simply, the Head-End) spectrum will be in the 450MHz-750MHz range, while in the upstream (from subscribers' homes to the Head-End), the smaller and more "difficult to manage" 5MHz-42MHz (up to 65MHz in Australia [WHIT 96]) portion of the spectrum will be used.
This is shown in Figure 3.7. It is expected that, in order to better cope with power limitations, cost and channel error characteristics, each of the upstream and downstream paths will be further subdivided into channels.

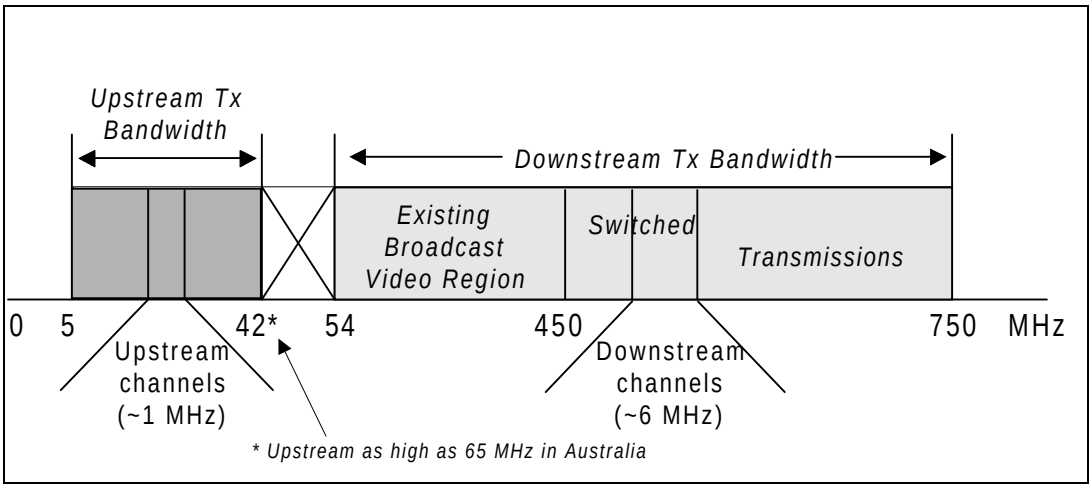


Figure 3.7: Upstream and Downstream Spectral Allocation in a two-way HFC Network

The downstream channels will be 6MHz wide, and possibly support data rates of 30-40Mbps (or more), while the upstream channels will be about 1-2MHz in width, and supporting about 2-10Mbps (or more) of digital data. The downstream channel is broadcast, while each upstream channel may be shared by up to a few thousand stations.

Two "domains" are identified in an HFC network - that of the fibre and that of the coax. The fibre domain extends from the Head-End to the single fibre node serving the customer's neighbourhood, containing anywhere between 500 and 2000 homes. A number of such neighbourhoods may be connected to the single Head-End. The coax domain covers the "last mile" of distribution to the customers. It may be expected that the distance from the farthest home to the Head-End will not exceed 80kms, while the coax domains (i.e. the neighbourhoods) will not exceed some 16km or so of cabling. This means that for any system size, the homes will be in general located only along the last 20% of the full HFC network distance.

It is clear that, being a shared physical medium system, with the potential of many thousands of users wishing to get on "the superhighway" at the same time, HFC networks will need sophisticated MAC protocols at the second OSI layer (i.e. the data link layer). One particular issue which needs to be addressed by any prospective protocols, and distinctly a characteristic of HFC networks, is that the tree and branch HFC network topology, that has grown over several decades of CATV network deployment, is ideally suited for broadcast transmissions. However, such a topology does pose a significant challenge for offering access in the upstream to many users simultaneously. Namely, stations cannot listen to each other's transmissions directly and therefore cannot self-regulate their access to the network. The result is that any candidate protocol operates on a very centralised philosophy, whereby the Head-End station plays a major role in co-ordinating both its own and each station's transmissions.

A number of candidate MAC protocols are compared and contrasted in Section 3.3.5, while a detailed description, analysis and simulation study of one such protocol (Fair - Centralised Priority Reservation) is undertaken in the next chapter, with substantial references to relevant parts of the Draft IEEE 802.14 standard, released in December 1996.

3.2.1.4.3 HFC vs. ADSL: How to Compare Between the Two Architectures ?

When comparing these two technologies, one thing quickly becomes apparent: the primary issues are those which directly impact on the cost of provisioning and of the infrastructure, as shown in Table 3.1. Most of the table's information was situated at the ADSL Forum Web-page [MAXW 97]:

Advantage of ADSL	*Advantage of HFC Networks*
(1) Present Numbers and Forecast Growth	
The number of telephone lines already installed world-wide that can support ADSL (≈400 million) far outweighs the number of "upgrade ready" or readily available HFC lines (≈6 million). Aggressive HFC upgrades will not improve that ratio (400:6) to better than 10 to 1 in the next five or six years.	
(2) Cost of Network Upgrade, Build and Operation	
The infrastructure costs required to either build new HFC networks or upgrade existing one-way CATV networks, overwhelm the fact that ADSL is not a shared network solution like HFC, and will hence be a higher network cost solution.	Offer a less expensive network solution because of the shared architecture, which, depending on the protocol used, acts as an almost ideal data concentrator (see Section below), in effect a distributed access node. This removes the need for any further concentration to make efficient utilisation of expensive router or ATM access device ports. There is a case study showing that the shared-resources approach of HFC systems realises huge cost savings over point-to-point ISDN connections per normalised peak bandwidth [GILL 95].
(3) Line Speed: Distance, Noise and User-number Limitations	
Unlike CDM speeds, ADSL line speed does not depend at all on ingress noise and the number of simultaneous users seeking access to a shared line. Therefore, a given ADSL user will be guaranteed a connection-long dedicated bandwidth in the low Mbit/s, unlike with CDMs where an increasing number of subscribers joining the line makes high speed upstream channels unattainable.	Due to signal-boosting amplifiers in the coaxial cable, CDM speeds do not depend at all on distance (in ADSL lines, a distance-bandwidth trade-off applies).

Advantage of ADSL	*Advantage of HFC Networks*
(4) Remote Fault Diagnosis and Prevention	
ADSL architectures currently enjoy an advantage over HFC systems, in this area. The highly distributed nature of the coaxial HFC network segment, and the possibility for a network fault located at some unknown upstream point to cause a fault to downstream users, makes it exceedingly difficult to quickly and accurately pinpoint the failure location. Furthermore, using ADSL's inbuilt forward-error-correction (FEC) and cyclic redundancy check (CRC) functionality, it is possible to detect or even pre-emptively react to impending individual line failures.	
(5) Permanent On-Line Connectivity	
The expected traffic multiplexing method in the local exchange (point of ADSL user convergence) is going to be something based on Ethernet (CSMA/CD) or on an ATM multiplexer. In either case, just as can be done in any standard office LAN, the ADSL users can be treated as permanently "on-line" user ports, which only use the traffic concentrator resources IF there is actually data to send or receive.	The nature of the MAC layer protocol operating over HFC networks means that it is possible, without wasting ANY network resources, for a user to be continuously on-line, 24 hours a day, 7 days a week. Network resources would only be used when the customer sends or receives information. In this area both the HFC and ADSL architecture provides no particular advantage over the other one.
(6) Possibility of "Graceful" Capacity Upgrade	
	In HFC systems, no change is required in the customer equipment when the time comes to upgrade the guaranteed bandwidth per customer. The fibre nodes serving multi-user neighbourhoods are brought closer and closer to residences, thus bringing down the number of users SHARING the coaxial cable segment, giving rise to the FTTC and FTTH architectures already described in Sections 3.2.1.1 and 3.2.1.2. With ADSL systems, the speeds already push the copper plant's physical capabilities to the limit, and no such incrementally graceful upgrade to FTTC / FTTH architectures would be possible, short of rolling out fibre to each customer.

Table 3.1: Primary Criteria for an ADSL vs. HFC Comparison

There is also a set of technical issues which are considered to be of a secondary nature, because they do not have the magnitude of impact on direct costs associated with each technology, and can hence be resolved with comparatively minor research and development efforts.

- **Security -** Since HFC systems involve broadcast transmissions in the downstream, intended or unintentional wire-tapping is a serious issue. In order to tap an ADSL line on the other hand, would involve an intrusive method (wires are often underground) as well as knowing the exact modem settings established during initialisation. It would thus seem that ADSL is inherently more secure than HFC systems [MAXW 97], and so encryption, while needed for both systems, is likely to be a more serious issue for HFC networks.

- **Reliability -** Whether a segment of coaxial cable in an HFC network is cut, or whether that same segment's streaming transmitter, or bidirectional amplifier malfunctions, **ALL USERS** on that segment will lose their service. ADSL modem failures only affect one user, and telephone lines are historically quite renowned for their reliability [MAXW 97]. In addition, the issue of network powering is more difficult to satisfactorily resolve for HFC systems, due to the number of directional taps and two-way amplifiers which need stable and reliable powering. Some of the CATV operators in the world have chosen to provide two to three hour battery back-up (with a view to the provision of HFC telephony services) but this is of course a very costly undertaking. In ADSL, the only difficulty would be loss of customer power (since local exchanges have existing battery or generator back up for extensive time periods) and even then, the loss of power to one region of a local exchange's coverage area, would not necessarily affect customers in other regions.

- **Quality of Service** - As [MAXW 96] and [MAXW 97] points out, CDM users suffer progressive degradation of quality of service with an increasing number of users becoming active on the shared line, due to increased ingress noise, increased channel contention and reduced reliability. While ADSL users will suffer no degradation based on traffic or numbers of users in the access network, all ADSL lines must at some point be multiplexed at the local exchange. There are certainly going to be peak times of day where the ADSL concentrator will encounter congestion. If the ADSL traffic concentrator's output speed is not greater than an equivalent CDM's speed, the congestion will result in identical loss of service quality for both types of access network. However, the remedy for problems of this nature in ADSL architectures is cheaper, because it just involves adding concentrator capacity, rather than physically splitting one HFC fibre serving area into two or more.

- **Home Wiring -** As ADSL will be offered over standard phone lines, it has the advantage of already being positioned conveniently close to today's typical Internet-connected home PC, which is located near a phone line. Cable modems will in most cases require new wiring (to lead the cable to the PC location(s)) inside the home, while many ADSL installations will not require new wiring, other than for the standard phone line splitter which can reside anywhere within the home.

- **Standards -** Cable television, as a business, has a very poor track record of consistent standards development and enforcement. It is thus likely that the fact that quite a few CDMs have been deployed well before the IEEE 802.14 standard has been finalised, will have possible detrimental ramifications on both the network and CPE interworking capabilities. To complicate matters even further, there are two alternative standards being developed by the Digital Audio Visual Council (DAVIC) and the

Cable Television Laboratories (CableLabs) which may not be compatible with the IEEE 802.14 standard in all areas when they are completed, because of their distinctly "lukewarm" support for the idea of ATM over HFC networks and intent on simply providing a broadband Internet service (i.e. IP over HFC). On the other hand, ADSL is a well standardised and scaleable technology where the five international standards organisations involved do not appear to be developing competing standards (American National Standards Institute - ANSI / T1E1.4, ADSL Forum, DAVIC, ATM Forum and European Telecommunications Standards Institute - ETSI).

While the above-mentioned "secondary issues" have largely favoured the ADSL technology over HFC systems, an important point needs to be raised here, and that is that the two types of access will start off in most areas of the world aimed at slightly different (but overlapping) segments of the market. Namely, the ADSL offering is ideally suited to the voice, video and data needs of a **range of business users** from corporate to small/home office. On the other hand, HFC networks are ideally suited to the **residential user** who is more likely to want a highly asymmetrical service, without any particular requirement for absolutely iron-clad guarantees of upstream bandwidth.

The markets for the two technologies certainly overlap: with sophisticated improvements for the IEEE 802.14 protocol under way, HFC systems will also be able to provide circuit emulation and other guaranteed bandwidth services, as well as being able to carry many ATM-based service types with quite different QoS. This will allow HFC to make inroads on the lucrative small to medium business market. The converse is also true. In areas of the world where no HFC cabling exists, and even in some where it does, ADSL will make significant inroads into the biggest market of all - the residential market.

3.2.1.5 Wireless Radio Architecture

One of the architectures envisioned to provide wireless access to users' residences is that shown in the FTTC scenario of Figure 3.3, and described in [MONT 95] as a "wireless PCS distribution system". The system essentially consists of fibre serving areas, catering for not more than 500 homes, each of which would be covered by a Base Station (BS), co-located with the fibre termination. The BS would be the point of signal conversion, from optical to radio. However, what is presented in Figure 3.3 is only one particular use of a wireless architecture, aimed mainly at residential access where user mobility is unimportant. A crucial additional functionality of radio access, is that it eliminates the inflexible paradigm of fixed location, and, provided that the user has a portable terminal, allows full user-terminal mobility within the bounds of the wireless radio network in question, [XIE 95], [RAYC 97].

Today's leading player in radio access is arguably wireless ATM (WATM), which is widely regarded as "the general solution for next-generation personal communication networks (PCN), that will be required to support multimedia", according to [XIE 95], [RAYC 97], [PASS 97] and references therein. Note that the term "next-generation PCN" includes both public PCS systems and private wireless local area networks of the future, [RAYC 92]. The authors of [NARA 97] point out that one of the key features of WATM is that it aims to provide seamless support of qualitatively similar multimedia services on both

fixed and mobile terminals. This concept is extremely significant, because it highlights the fact that a WATM-based personal communications network is not merely yet another access network architecture, but is a technology providing a unifying interface between heterogeneous network infrastructures (i.e. fixed and radio), and one which has the potential to enable universal user-terminal mobility, that being the capability to move seamlessly from a fixed network to a wireless network, and vice-versa.

The work in [RAYC 92] presents some of the basic architectural details of a WATM-based wireless network:

- a desired coverage area (say a city suburb or town) is served by a number of PCS micro- and pico-cells, each of which is between 500 and 100 metres in radius.

- the cells are interconnected by a fibre-based PCS interconnection network.

- the area of each individual cell is covered by its base station, which consists of high-speed radio modems providing reasonably reliable transmission in the 5 GHz U-NII band in the US and the HIPERLAN band in Europe. Operation in higher frequency bands, for example 20-30 GHz or 60 GHz, is currently under consideration.

- the expected operating frequency range of WATM transceivers is between 20 and 25 MHz, with typical bit rates for the radio physical layer at around 25 Mbps, with a goal of supporting per-virtual-connection (i.e. per-VC) service bit rates of around 6 Mbps sustained and 10 Mbps peak.

- a key requirement for WATM radio modems is to be able to support burst operation, with relatively short preambles which allow high-speed transmission of short control packets and ATM cells.

One of the key issues with WATM is that, in interfacing wireline and wireless networks, it must provide wireless-specific extensions to the pre-existing ATM protocol stack [NARA 97]. In other words, the original intention in the design of ATM was that it would operate in a fibre-based broadband network environment, characterised by extremely high levels of reliability and dedicated access to the transmission medium. Unfortunately, a radio network provides none of these traits: it is both prone to a high level of noise and other errors, and it is inherently a shared, not dedicated, transmission medium. Therefore, one of the primary additions to the ATM protocol stack in order to provide WATM functionality, has been the inclusion of a wireless-specific medium access protocol. The similarity of this WATM MAC protocol to other MAC protocols we will study is of great interest to us, and is discussed in Section 3.3.6.

3.2.2 The Ideal Access Network: An Ideal ATM Multiplexer

One of the primary contributions of [SLOS 96] has been to propose an ideal ATM multiplexer as a model of a Time Division Multiplexed (TDM) B-ISDN access network. The multiplexer is said to be ideal in the sense that it is both *fair* (see definition below) and *work conserving* (meaning that the output link, or the server, never stands idle if there is data to transmit). The set of N subscribers are connected in a star configuration to an ATM multiplexer unit. The multiplexer superimposes the traffic originating from the

N sources at ① (see Figure 3.8.), resulting in the Queue Input Process (QIP) at ②, and as defined in [SLOS 96]. The QIP is fed into a common waiting queue at ②, where concurrent cell arrivals are stored in random order (and we thus define *fairness*). The output of this queue ③ is then directed upstream to the head-end or local exchange (depending on the type of network).

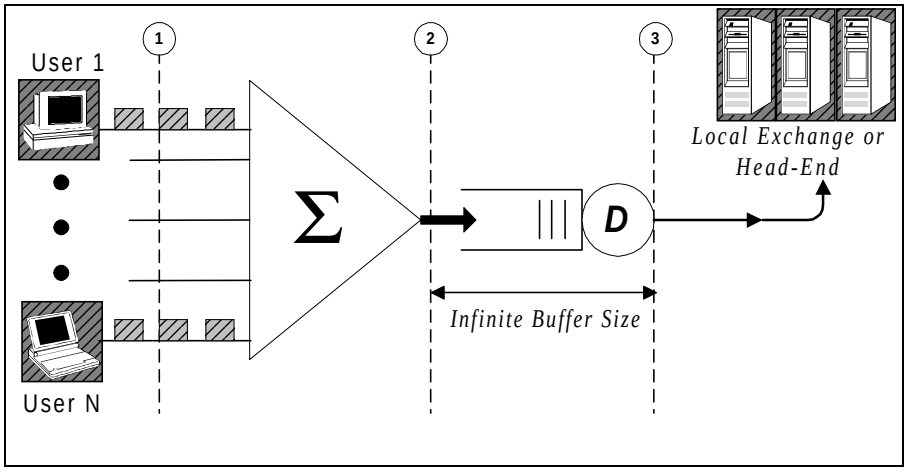


Figure 3.8: Model of the Ideal Access Network - An ATM Multiplexer

Lindley's equation governing the number of cells in this discrete time ATM multiplexer is given by [SLOS 94]:

$$n_k = [n_{k-1} - 1]^+ + a_k \qquad (3.1)$$

where n_k represents the number of cells in the system at time slot k, a_k the number of cell arrivals at time slot k, which satisfies $0 \le a_k \le N$, for any k, and $[\cdot]^+$ represents the function $\max[0,\cdot]$.

[SLOS 96] selects the discrete-time ideal ATM multiplexer as the benchmark TDM access network model, because of the following features:

- No fixed transmission delay is introduced to each cell's transmit time.
- The queue is served using the FIFO (First In, First Out) discipline, the cell service policy which introduces the smallest delay variance.
- The random delay in the queue is purely caused by temporary congestion, rather than design limitations or system imperfections.
- All input "inlets" are handled equally, and thus it is a totally fair access network.

In carrying out a detailed analysis of one candidate MAC protocol designed to run over HFC access networks later on in this work, we use this same ATM ideal multiplexer (IM) to the one described above, to serve as an ideal TDM access network model (i.e. a delay and throughput performance benchmark).

One important issue in the “battle” for dominance in the access network market, between the two major competing technologies of ADSL and CDMs, is the efficiency of aggregation of all traffic generated by and directed to the customers. This is something which is of utmost importance to the network operators, because it allows both the customer and network operator to benefit from the same economy of scale that large institutions have long enjoyed in the form of LANs and WANs, [BISD 96b]. As mentioned in Section 3.2.1.4, the fact that ADSL is a point-to-point system and requires equipment and network resources to be permanently dedicated to each subscriber, already has major provisioning cost implications. In addition, and intimately related to the direct cost of providing dedicated infrastructure, is the issue of how early on in the access network, and to what extent, does successful multiplexing of traffic occur, in the two competing network architectures?

We have already stated that the preferred multiplexing benchmark is the IM, and have described it above. Of interest now, is how close to the IM are these two access network technologies, and how early in the access network may they be modelled by the IM?

3.2.2.1 HFC Networks - CDM Technology

As will be shown in Section 1, the performance of the particular MAC protocol which we study in detail, *is almost identical to that of an IM*, under a diverse range of system parameters and types of carried traffic. It is also made clear in the same section that the protocol under consideration highly resembles the Draft IEEE 802.14 standard, hence it can be taken to be accurately representative of future HFC access networks’ traffic multiplexing performance. Finally, the distributed tree-branch nature of HFC networks gives rise to something akin to a very large-scale distributed IM, with contending nodes dotted along the shared infrastructure. This means that the multiplexing of traffic occurs immediately - out there, in the actual access network; and that by the time data reaches the Head-End, we have an almost ideally multiplexed stream of cells, frames (or other types of MAC-layer payload data units).

3.2.2.2 ADSL Technology

On the other hand, ADSL modems need to be placed both at the customer site, and at the local exchange, thus giving rise to one pair for each subscriber. The first thing to notice about this architecture is that any aggregation, no matter how efficient it is, can only occur at the first instance where the multiple ADSL lines converge (i.e. at the local exchange, for first-generation ADSL networks). Unlike with HFC systems, this point is much “deeper” into the provider’s network.

The second issue with aggregation of traffic streaming into the local exchange from many ADSL lines, is to do with the nature of the multiplexing process itself. Namely, in what way is a multi-line ADSL multiplexer going to function? As was explained in Section 3.2.1.3.3, the answer is going to be dependent on the vendor, and on what protocol the vendor’s ADSL equipment is designed to support at the higher layers. Clearly the first, Ethernet-based, type of traffic aggregation system does not behave according to an IM benchmark; nor is it designed to do so, in light of the connectionless Internet Protocol (IP) traffic

which is likely to be carried in this arrangement. The second solution does involve an ATM multiplexer unit, but due to some vendors' ways of implementing such a device, it is far from a foregone conclusion that such a unit will behave identically to an IM; or for that matter, any closer to an IM than the HFC networks do.

The importance of a technology being able to emulate an IM lies in the IM being best able to satisfy stringently low delay and delay variation requirements for the many multipriority, high Quality of Service (QoS) applications which either exist today or are planned in the future, not the least important of which is voice-grade circuit emulation and real-time video conferencing. When considering which technology scores more points in terms of (i) being closer to our ideal benchmark, more consistently and independent of vendor implementation; and (ii) achieving a multiplexing gain as close as possible to the end user, it would appear that HFC networks currently have the upper hand. This assessment in no way negates the many other areas in which ADSL technology is preferable, as outlined in Section 3.2.1.4.3.

3.3 MAC Protocols for LAN, MAN, HFC and Wireless Architectures

3.3.1 Classical Slotted Aloha

We begin our review of the various MAC protocols by first considering a popular multiaccess approach, which has found its way in some shape or form, into many modern medium access protocols. We refer to Slotted Aloha [ROBE 72], the improved version of the original Aloha network [ABRA 70]. The original Aloha was developed by Abramson around 1970, to provide radio communication between the central computer and various data terminals at campuses of the University of Hawaii. We restrict our attention to the slotted improvement of the original algorithm, because not only does it give twice the signalling throughput, but its slotted time paradigm also fits in with many of today's access technologies (such as HFC or wireless access networks, and even LANs). As we shall see from the survey of the various MAC protocols that follows, the conceptual simplicity and ease of implementation of the Slotted Aloha approach has made it a very strong contender for any contention-based components of MAC protocols in HFC or WATM systems.

In this section we firstly define and discuss the assumption set of the classical Slotted Aloha model: Section 3.3.1.1 examines the ideal slotted model, while Section 3.3.1.2 looks as the Aloha algorithm itself. The final part of our study of classical Slotted Aloha, Section 3.3.1.3, is arguably the most significant, since it explores the concept of instability (i.e. infinite delay in accessing the shared medium) and methods of making Slotted Aloha practically usable, either through formal stabilisation techniques or by operation within specified load bounds.

3.3.1.1 The Ideal Slotted Multiaccess System

Let us first describe the classical *Idealised Slotted Multiaccess Model*, given in [BERT 92] and some of the references therein, and discuss some of the model's assumptions. The system topology is that shown in Figure 3.9, with *m* transmitting nodes and one receiver, which we can think of as a central controller. The nodes cannot hear each other, and all communication goes through the central controller - a topology reminiscent of the HFC, wireless and satellite network architectures we have discussed earlier in this chapter.

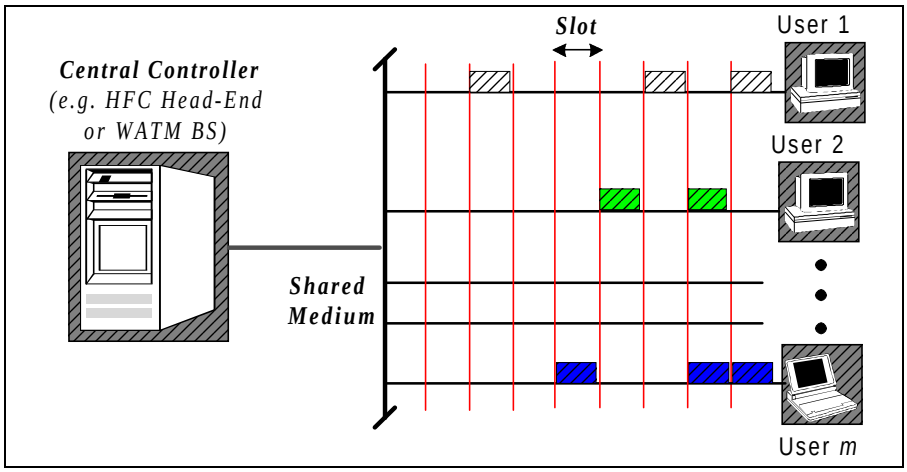


Figure 3.9: One Receiver - *m* Transmitters Topology, Idealised Multiaccess Model

The six cornerstone assumptions listed by Bertsekas and Gallager in [BERT 92] are:

(1) ***Slotted system***: Transmitted packets have the same length, and each packet requires one time unit, called a *slot*, for transmission. All transmitters must be synchronised to ensure that the reception at the single receiver node starts at an integer slot (measuring time) and ends before the next integer slot.

(2) ***Poisson Arrivals***: The packets arrive for transmission at each of the *m* transmitting nodes, according to independent Poisson processes, so that λ is the overall arrival rate to the system, and λ / m is the arrival rate at each of the nodes.

(3) ***Collision or Perfect Reception*:** If two or more nodes transmit within the same slot, then there is a collision and the signal is garbled so that the central controller gets no information about the contents or source of the transmitted packets, or how many of them were present in that slot. Also, the possibility of channel errors is neglected, so that if one node transmits in a given slot, perfect reception of the packet at the central controller is assumed.

(4) ***{ Idle, Success or Collision } Immediate feedback:*** At the end of each slot, the central controller provides each and every node (not just the active ones) with feedback, specifying whether no packets, one packet or more than one packet (and hence a collision) were transmitted in that slot.

(5) ***Retransmission of collisions:*** Each packet involved in a collision must be retransmitted in some later slot, with this continuing until the packet is successfully received. Nodes with a packet that must be retransmitted are called *backlogged* nodes.

(6a) ***No buffering****:* If one packet at a station is currently waiting for transmisison or is in the process of colliding, new arrivals at that station are discarded and never transmitted. The alternative to this assumption is (6b).

(6b) ***Infinite set of nodes*** ($m = \infty$): The system has an infinite set of nodes and each newly arriving packet arrives at a new node, so that packet discard never happens.

The effect of assumption (1) is twofold: it simplifies analysis by defining a discrete-time system, and it precludes the use of carrier sensing (CS) or early collision detection (CD) techniques (both of these techniques improve the utilisation efficiency of the multiaccess channel). Models involving retransmission of flawed messages in CSMA and CSMA/CD systems based on Poisson and Batch Poisson arrivals have been explored in depth in Kleinrock and Tobagi [KLEI 75], and Heyman [HEYM 82], and references therein. A final note regarding assumption (1) is that, although not simple, it is possible in practical systems to synchronise the transmitters for slotted arrival at the central controller. This is accomplished through a small amount of feedback from the central controller, relatively stable clocks, and ample guard time at inter-slot boundaries.

Assumption (2), stating that arrivals are modelled by Poisson processes, enables analytical tractability, but is wholly unrealistic for real traffic. Likewise, real systems will definitely be prone to errors due to noise, making assumption (3) seem very idealised. A more realistic Slotted Aloha-based model, both in terms of capturing the possibility of errors and extremely correlated traffic (with memory), is the subject of Chapter 4.

While assumption (4) may appear quite unrealistic in the case of very high-speed systems or those with very long propagation delay (e.g. satellite networks are a prime example), it appears to be justified for most realistic HFC and wireless systems by the fact that the slot size is sufficiently large to allow stations to react immediately to feedback information. It would not be uncommon for an HFC or wireless system to have a slot size large enough to permit the central controller to read the contents of the upstream channel, make a decision on the contention outcome, and in the very next downstream slot give feedback to the nodes. This assumption of round-trip delay being smaller than slot size is also made in related contention resolution algorithm studies for HFC systems [BISD 96c], [LAUB 95], [SALA 96c].

We stress here that the nodes themselves do not have enough information to deduce feedback information - it must be externally supplied, by the controller. Most realistic transmission speeds, propagation and central controller processing delays, would suggest that at worst the nodes would receive feedback with a single slot delay. This statement does make the assumption however, that nodes in the HFC and wireless systems are capable of synchronising to simultaneously receive (from the downstream channel) and

transmit (on the upstream channel). The final word on feedback delay is quite a promising one: as [BERT 92] points out, although delayed feedback complicates multiaccess algorithms (in proportion to the level of extra delay), it causes no fundamental problems in operation or analysis.

There is no question that assumption (5) is reasonable in providing a reliable communication path; it is desired that collided packets do eventually reach their goal. In this respect, assumption (6a) does not make a lot of sense since it permits a backlogged node to simply discard any new traffic. However, assumption (6a) is intended to provide us with a *lower bound to the access delay* for a wide variety of systems with buffering and flow control. Assumption (6b) on the other hand, compliments (6a) since it provides us with an *upper bound to the access delay* that can be achieved with a finite number of nodes. As [BERT 92] explains, if the performance using assumption (6a) is similar to that using (6b), it is reasonable to expect that we have a good approximation to the performance of a system with arbitrary buffering assumptions.

3.3.1.2 The Slotted Aloha Algorithm

Having established the framework within which the multiaccess algorithm is to function, we now turn to the core principles of Aloha itself. The basic idea is that each unbacklogged node simply transmits a newly arriving packet at the first slot boundary after its arrival. In this way, the node risks occasional collisions, but achieves exceedingly small delay if collisions are rare enough. To put this behaviour in context, consider classic time division multiplexing (TDM) where each of the m nodes would have a preassigned slot, eliminating the possibility of collisions but only at the expense of a larger access delay (m / 2 on average). Continuing our description of Slotted Aloha, when a collision does occur, each node sending one of the colliding packets, discovers the collision at the end of the slot and becomes backlogged. Note that if every node were to retry with certainty in the slot following a collision, a subsequent collision would be guaranteed. The way to resolve the collision is for each backlogged node to wait a random number of slots, before reattempting transmission of the collided packet. As [BERT 92] points out, many different probability distributions have been proposed in order to exactly quantify this random number of slots. Note that these probability distributions may be thought of as the main part of the contention resolution algorithm, or CRA.

For example, the original version of Slotted Aloha, as proposed in [ROBE 72] employed a CRA with a *uniform* distribution between retransmissions. Another CRA proposal, especially for radio channels where only limited feedback (collision, successful transmission) is available to the nodes, is the *binary backoff* algorithm [METC 76], where if a packet has been transmitted unsuccessfully i times, then for each of the ensuing slots the probability of retransmission is fixed at $p = 2^{-i}$. This is effectively a geometric distribution with an increasing mean time to success, as the number of collisions increases. Another (very similar) way of implementing the binary backoff algorithm is described in [METC 76]: uniformly distribute the probability of retransmission over the next 2^i slots after the i^{th} failure. The third well-known CRA is commonly referred to as p-persistence [BERT 92], since after a collision each node “persists” in retransmitting, but only with a probability p in each following slot. This means that the number of slots

from a collision until a given node involved in the collision retransmits, is a *geometric* random variable having value $i \geq 1$ with probability $p(1-p)^{i-1}$.

Focusing our attention on the p-persistence CRA, let us define P_{succ} as the probability of successful transmission in a slot; since only a single transmission per slot may be considered successful, P_{succ} also quantifies the expected number of successful transmissions per slot, and hence is a measure of the throughput of Slotted Aloha employing the p-persistence CRA, and under assumptions (1) through to (6) discussed earlier. [BERT 92] uses simple discrete-time Markov chain analysis to obtain the approximate expression for P_{succ} as,

$$P_{succ} \approx G(n)e^{-G(n)} \tag{3.2}$$

where *G(n)* is the attempt rate when the system is in state *n* (i.e. when *n* nodes are backlogged). *G(n)* is the number of attempted transmissions in a slot, and is given by the expected number of arrivals plus retries per slot, so that,

$$G(n) = \begin{cases} (m-n)\cdot(1-e^{-\lambda/m}) + np, & for \quad finite \quad m \\ \lambda + np, & for \quad m = \infty \end{cases} \tag{3.3}$$

Recall that in equation (3.3), a finite number of stations corresponds to the no buffering assumption, (6a), while infinite *m* corresponds to assumption (6b). Equation (3.2) tells us that the maximum achievable throughput of this classical Slotted Aloha model is limited to $1/e \approx 0.368$ (if *m* is very large, or infinite). Next we turn to examine the operational dynamics of a Slotted Aloha system (i.e. defining equilibrium points), and explore the issue of stability.

3.3.1.3 Slotted Aloha Dynamics and Instability

A Slotted Aloha system has interesting properties when the number of nodes *m* is very large. For example, choosing *p* to be moderately large, we aim to avoid large delays after collisions. If λ is small and not many packets are involved in collisions, this works well and retransmissions are normally successful. However, if there is a run of "bad luck" with reattempted collisions, and the number of backlogged nodes gets sufficiently large to satisfy $pn >> 1$, then collisions occur in almost all the slots thereafter and the system remains backlogged for a long time, if *m* is finite. Even worse, the system remains backlogged for an infinitely long time if $pn >> 1$, and under assumption (6b) $m = \infty$.

In order to gain quantitative insight into the way in which the state, *n*, of the Slotted Aloha system fluctuates, let us define the *drift* in state *n*, *D(n)*, as the expected change in backlog over one slot time, starting in state *n*, keeping to the same notation of [BERT 92] for consistency. Therefore, *D(n)* is the expected number of new arrivals accepted into the system (equal to $(m-n)\cdot(1-e^{-\lambda/m})$ for a finite *m* system, and λ for an infinite *m* system) minus the expected number of successful transmissions during that slot (the number of departures),

$$D(n) = \begin{cases} (m-n)\cdot(1-e^{-\lambda/m}) - P_{succ}, & for \quad finite \quad m \\ \lambda - P_{succ}, & for \quad m = \infty \end{cases}, \qquad (3.4)$$

with P_{succ} given by equation (3.2). We now use Figure 4.4 from [BERT 92] to illustrate the fluctuating behaviour of Slotted Aloha. Namely, in Figure 3.10, we present a graph for both finite and infinite number of nodes, which depicts the arrival and departure rates on a horizontal axis representing both the state *n*, and the attempt rate, *G(n)*, as labelled.

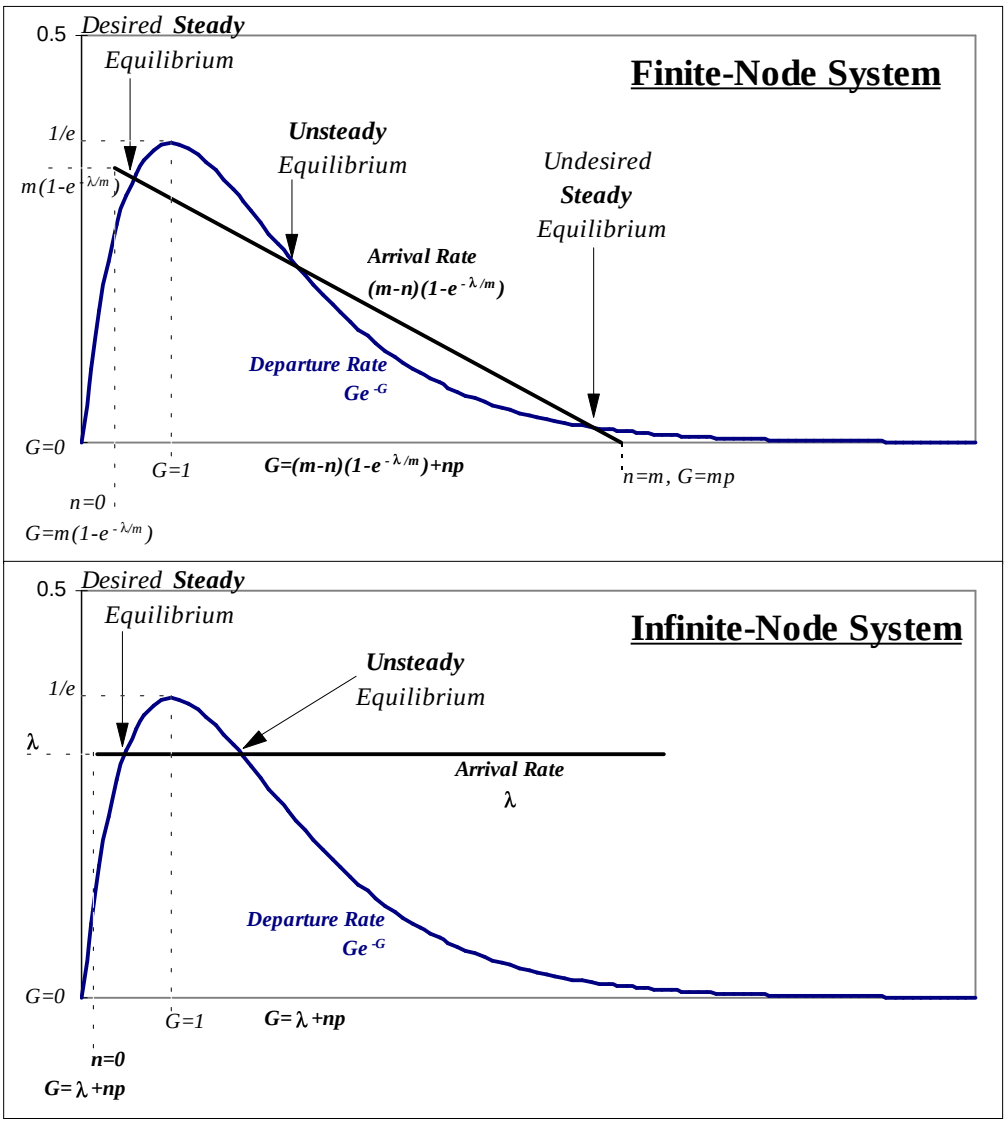


Figure 3.10: System Dynamics of Finite- and Infinite-Node Slotted Aloha

Figure 3.10 allows us to visualise equations (3.2) and (3.4), for the case $p > (1-e^{-\lambda/m})$, which is really the only case of interest, since a smaller retry rate would frequently cause large build-up in the system state *n*, and excessive access delays per packet. The drift, *D(n)*, is the difference between the line and the curve, and when it is zero we say that as *equilibrium point* is reached, such that the arrival and departure rates are equal.

Turning our attention first to the upper part of Figure 3.10, the finite *m* case, we note that three points of equilibrium exist, with an *unsteady* equilibrium straddling two *steady* equilibria. By unsteady, we mean that the state drift near this equilibrium point is such that, as soon as the system slightly moves away from this point, it will fluctuate further and further away from the unsteady equilibrium, due to the growing imbalance between the arrival and departure rates. By contrast, the drift in the vicinity of the two steady equilibria pushes the system state back towards the equilibrium point after any fluctuation, and so during operation the Slotted Aloha system tends to cluster around the two steady equilibrium points, with rare excursions between the two. It is important to note that only the leftmost steady equilibrium point is *desired*, because the throughput (departure rate) of the steady point on the right is almost zero.

Turning now to the lower part of Figure 3.10, which represents the dynamics of an infinite node Slotted Aloha system, we note a very similar situation, except that the arrival rate is equal to λ, and is completely independent of system state, *n*. Hence, it is manifested as a horizontal line. It is important to note that the rightmost steady equilibrium disappears in this system. Once the state of the system passes the unsteady equilibrium, the drift given by equation (3.4) is increasingly positive, causing the number of backlogged nodes to increase without bound. [BERT 92] emphasises that in this case the corresponding infinite case Markov chain does not have a steady state distribution.

We conclude our discussion by explaining the concept of **stability**. According to [BERT 92], *"a system is defined to be stable for a given arrival rate, if the expected delay per packet (either as time average or an ensemble average in the limit of increasing time) is finite"*. With this definition in mind, we come to fully appreciate the main difference between systems described by assumptions (6a) and (6b). That is, a system with a finite number of nodes *m*, under assumption (6a), is always stable under this definition, regardless of λ. Even though the delay may be exceedingly long with some choice of parameters, it is nonetheless always finite. On the other hand, for any arrival rate λ, greater than zero, an infinite-node system, as per assumption (6b), is always unstable.

We can now appreciate that both finite- and infinite-node Slotted Aloha systems have problems with steady operation. While a finite-node system is always stable with finite delay, if system state passes the unsteady equilibrium, it proceeds to a region of operation where the throughput is almost zero with intolerably high delay per packet. For all practical purposes, this mode of operation is equivalent to instability. And in fact true instability is exactly what happens to an infinite-node system, when it exceeds its unsteady equilibrium. This is why ordinary Slotted Aloha is said to be an unstable contention resolution algorithm. In fact, Aldus [ALDU 91] has conjectured that there can be no acknowledgement-only based CRA such as Slotted Aloha, that is inherently stable and without the need for more complex centralised processes to effect stabilisation. This conjecture has not yet been rigorously proven. Over the years many *methods of stabilisation* for p-persistence Slotted Aloha have been proposed. We only mention the two most popular:

(a) Quasi-Stabilisation: Small *p* and Backlog Clearance

Considerable theoretical attention has been given to techniques whereby *p* is chosen to be small enough to keep both finite-node and infinite-node systems close to the desired steady equilibrium point, with near

certainty, [TSYB 85] and references therein. From a practical standpoint, especially for arrival rate λ smaller than 1/*e*, the move past the unsteady equilibrium would be extremely unlikely. And in this event, the system could be restarted with backlogged packets lost.

(b) True Stabilisation: Rivest's Pseudo-Bayesian Algorithm

A much better, and particularly simple and effective way to stabilise Slotted Aloha, is the Pseudo-Bayesian approach proposed by Rivest [RIVE 85]. An earlier, independently derived algorithm by Mikhailov [MIKH 79] is essentially the same, but Rivest's Bayesian interpretation does simplify understanding. The crux of the algorithm is in adjusting the value of *p* to its well known optimal value of 1/*n* (when in state *n*), at the beginning of every slot. It is envisaged that this would be the function of a central controller for HFC and WATM systems, and the value of *p* would be fed back to the stations in every slot. The Rivest algorithm treats all new packets as backlogged immediately upon arrival, so that the attempt rate becomes simply $G(n) = np$, when in state *n*, and the probability of successful transmission (departure rate) becomes $np(1-p)^{n-1}$. If we were dealing with unstabilised Slotted Aloha, one would think that this modification makes no sense, since *p* has to be small for operation close to the steady equilibrium, and thus all new arrivals would be unnecessarily delayed. However, as we shall see, the p-persistence parameter can be as large as 1 in this stabilised version of Slotted Aloha, provided the backlog is close to one. At the beginning of each slot, the (central controller resident) algorithm updates its estimate of system state, denoted by $\hat{n}$, depending on the outcome in the previous slot. So at the beginning of slot $k+1$, the estimate would be changed according to the equation,

$$\hat{n}_{k+1} = \begin{cases} \max\{\lambda, \hat{n}_k + \lambda - 1\}, & \textit{for idle or success} \\ \hat{n}_k + \lambda + (e-2)^{-1}, & \textit{for collision} \end{cases}, \qquad (3.5)$$

Having calculated the new $\hat{n}$, each node then updates its p-persistence parameter according to the rule,

$$p(\hat{n}) = \min\{1, 1/\hat{n}\}, \qquad (3.6)$$

which effectively limits the p-persistence probability to 1, since the $\hat{n}$ estimator may sometimes be less than one. A deeper investigation of how Rivest's algorithm was derived is provided in [BERT 92], but for our purposes, it suffices to say that it yields an improved, stabilised Slotted Aloha system operating close to its peak throughput of 1/*e*, by virtue of a very simple series of algorithmic modifications.

3.3.2 Survey of Popular Contention Resolution Algorithms (CRAs)

Many factors in the design of a MAC protocol have a direct impact on the collision probability experienced within the access network (see Section 3.3.5, which focuses in particular on HFC networks). However, once a collision has already occurred, we wish to define efficient retransmission rules, which will hopefully yield a resolution of the collision in as short an interval as possible. The set of such rules is

called the contention resolution algorithm (CRA), and is the topic of this section. In describing the various CRAs, we use the idealised slotted multiaccess foundation and its associated assumptions, as presented in Section 3.3.1.1. In the previous section, we mentioned three CRA variants, all belonging to the Slotted Aloha family of algorithms: uniformly distributed retransmission period, p-persistence, and truncated binary exponential back-off.

While each of these CRAs is a slightly different "flavour" of Slotted Aloha, they all nonetheless share the same random access *philosophy of individualism* [BISD 96c]. This philosophy is one where the network stations should only be interested in the outcome of their own transmissions, with this outcome being conveyed to them via positive acknowledgements, also known as feedback. Therefore, a typical Slotted Aloha station need only monitor the channel immediately after its transmission. The exception to this rule is Ethernet, discussed in the following section, where carrier sensing is used and the channel is monitored before transmission also.

The other major random access philosophy is that of *collective operation*, belonging to the *splitting* family of algorithms. The guiding principle here is that of a collective effort by all stations registered on the system to continuously monitor the feedback information provided by the central controller node, and update their state accordingly. Note that there are some variants of this family of CRAs, where only active stations need to monitor the feedback information (i.e. those currently participating in a collision or not yet allowed to participate but wanting to transmit data). In either type of system, all (re)transmission decisions are taken based on the current feedback and state information. Two main categories of splitting algorithms exist [BISD 96c]:

- *tree-search* - also known as blocked-access splitting algorithms since new packets are not allowed into the system while a collision resolution is under way. Further subdivision of these algorithms is possible, according to the first transmission rule used. This rule controls when and how the blocked stations are first allowed to transmit in the system.

- *stack* - also known as free-access algorithms since new packets are permitted to enter the system during an on-going collision resolution.

In general, both splitting and Slotted Aloha families may be separated based on the feedback they require for their operation. Feedback, and other important CRA classifiers applicable to both the Slotted Aloha and Splitting algorithm families, are explained in full below (extracted from [SALA 95], [BISD 96c]):

- **Sensing** refers to the way in which the CRA imposes the algorithm rules to the user population. A system referred to as a *full sensing* (FS) CRA enforces rules and restrictions on all users, even if they are not currently involved in the contention process. In such a case, all stations need to continually be monitoring the reverse control path. An example of FS CRAs are all blocked-access (tree) algorithms, since a new station is restricted to transmitting only in certain time intervals. A *limited sensing* (LS) CRA however, only requires those stations which are currently participating in unresolved collisions,

to process the Head-End feedback information. Free-access (stack) algorithms are an example of LS CRAs, in that all new arrivals immediately join the collision resolution process, without any signalling or restrictions. LS has two major advantages over FS: a significant saving in station computation power and better **robustness** (as qualitatively graded below in Table 3.2) to loss of information from the Head-End. The latter point refers to the fact that in FS, because the users do not rely on their own local states, they are totally dependent on communicated global state "updates". In the presence of errors on the downstream channel, the loss of this global state picture would cause a deadlock situation due to the loss of synchronisation (until a centralised recovery scheme took hold of the system problems).

- A CRA's **feedback** parameter describes the number of states which can be determined by a user from reading central controller feedback on the channel. Algorithms requiring three states (collision, success, idle) are those with *Ternary Feedback*, while algorithms where only the collision and success states are required, are known as *Binary Feedback* CRAs. Although the former type of scheme achieves greater throughput, in some access network types such as wireless environments, it is not possible to implement it.

- The **maximum stable throughput** of a scheme is the theoretical maximum value of arrival rate for which the system's departure rate is the same (i.e. for which the system remains stable). Another definition is that it is the maximum proportion of time during which the channel is used to make successful transmissions.

- The **Contention Resolution Interval (CRI)** is the time period during which (i) collisions occur, followed by (ii) retransmissions and ultimately (iii) overall contention resolution. The CRA has as its chief task to define exactly the rules of retransmission during this interval. When this interval has some predetermined maximum duration, it is said to be *bounded*. Tree-search splitting algorithms have the advantage of bounded CRIs; for all Slotted Aloha algorithms, it is not possible to identify such distinct intervals, so it is said that they have unbounded CRIs.

- CRA **stability**, using the earlier definition in this chapter, is used to describe whether or not the algorithm causes the system to become unable to cope at extreme loads, with the surge of request traffic. That is, if the *average access delay per packet is still finite* at a given arrival rate, the CRA is considered stable at that arrival rate. If the CRA is considered unstable, a secondary mechanism is required for acceptable system performance. Good examples of this secondary mechanism are a station's own load control mechanism, or, the addition of a second, usually collision free, signalling channel (an example is the piggybacked data request used in many proposed HFC MAC protocols, as we shall see later in this chapter).

- Ability to operate in a **DPD** environment, states whether the CRA can operate as part of a system where differences in propagation delay, are tolerated. This has an impact on CRA complexity.

Table 3.2 is a combined reproduction of Table 3 in [SALA 95], and Table 1 in [BISD 96c]. It summarises key performance and operational characteristics of six common CRA types, with more detailed information available from the references included in the leftmost column, or references provided from the documents where the table information is sourced from.

Characteristic ⇨ CRA ⇩	**Sensing**	**Feedback**	**Maximum Stable Throughput**	**Bounded CRI**	**Robustness**	**DPD**
***Unstabilised* Slotted Aloha (p-persistence)**	Limited	Binary	0	NO	Excellent	YES
Static Binary Tree-search [CAPE 79]	Full	Binary	0.346	YES	Poor	NO
Stack *(n=2)*	Full	Binary	0.360	YES	Poor	NO
***Stabilised* Slotted Aloha (p-persistence)**	Full **or** Limited	Binary **or** Ternary	0.368	NO	Excellent	YES
Stack *(n=3)*	Limited	Binary	0.402	NO	Good	NO
Dynamic Tree-search [CAPE 79]	Full	Ternary	0.429	YES	Poor	NO

Table 3.2: The Most Well-known CRA Types

Another important aspect of CRA performance is resiliency in the face of extreme stress scenarios. One such scenario is the simultaneous arrival of a large number of packets into a previously empty system [BISD 96c]. Of interest in this case, is the average number of slots it will take to clear (i.e. successfully transmit) all the packets, under the assumption that no new packet enters the system. Table 3.3 is reproduced from [BISD 96c], and it shows the average time to clear a batch of 2000 packets under the circumstances just described, for some of the well-known CRAs we have discussed previously:

Contention Resolution Algorithm	**Avg. Time to Clear N = 2000 packets (in slots)**	**$\alpha = L_N / N$**
Dynamic Tree-search (Ternary feedback)	≈ 5,328	≈ 2.664
***Ideal Stabilised* Slotted Aloha (p-persistence)**	≈ 5,436	≈ 2.718
Stack *(n=3)* (Binary feedback)	≈ 5,462	≈ 2.731
Static Binary Tree-search (Binary feedback)	≈ 5,770	≈ 2.885
***Implementable Stabilised* Slotted Aloha (p-persistence)**	≈ 6,400	≈ 3.2
***Unstabilised* Slotted Aloha (p-persistence)**	>10,000	>5

Table 3.3: Performance of Selected CRAs under Stress Conditions

Note that the quantity α may be thought of as the average contribution of each station, to the overall CRI length in slots, L_N. It is interesting to see that the ideal stabilised Slotted Aloha CRA is a very close second to the Static Binary Tree-seat algorithm. Interestingly in [BISD 96c] it is stated that the *ideal* stabilised Slotted Aloha is also non-implementable (recall from Section 3.3.1.3 that all stabilisation schemes are only estimation based and cannot guarantee peak theoretical throughput 100% of the time).

We note that the practically implementable stable Slotted Aloha CRA comes a distant second-last. This type of CRA stress testing will be used in Chapter 4 as a foundation for our suite of Slotted Aloha deadlock models, which investigate signalling channel resiliency to transmission errors and extreme inter-station correlation.

3.3.3 Local Area Networks: the CSMA/CD Ethernet protocol (IEEE 802.3)

One of the first and most detailed papers describing an Ethernet local network and its performance was by Shoch and Hupp, of the Xerox Research Centre [SHOC 80]. The paper was based on work in the late seventies and was written at a time when the concept of the modern-day Ethernet Local Area Network (LAN) was still a novelty. The paper describes the Ethernet communications network as a broadcast, multiaccess system for local computer networking, using the techniques of *carrier sense (CS)* and *collision detection (CD).* It is a system with a logical multiaccess bus architecture, where each station sees all passing traffic but only reads packets intended for its identification number. Conversely, each station broadcasts each packet on the bus, for all other stations to see.

Ethernet is an enhanced member of the Slotted Aloha family of random access algorithms: when an Ethernet station initially tries to send a new packet, the CS mechanism is used in order to force a deferral of transmission if another transmission is in progress, so that a collision is avoided. Due to propagation delay, it is possible that two or more stations sense that the channel is idle and begin transmissions simultaneously, producing a collision. However, because each sender continues to monitor the channel in order to make sure that its signal is not being "overwritten" with another, collision detection is made possible. In such a case, both stations would realise the occurrence of the collision and would stop transmitting. In order to prevent repeated collisions, each collided station waits for a random amount of time before retransmitting. To avoid overloading the channel, the range of the retransmission interval is increased at times of heavy load, using the *truncated binary exponential backoff algorithm* mentioned in Section 3.3.1. The fact that there are millions of PCs attached to Ethernet LANs world-wide, makes this particular (enhanced) variant of Slotted Aloha the most popular and widespread random access approach at the present.

A station which has made n retries (i.e. a collision lies behind every retry), will choose at random a wait-until-retry time from an interval given by,

$$[1, \quad 2^{n+1} - 1] \tag{3.7}$$

There is a limit to the number of retries, n, which may be made by a station, and in most Ethernet implementations the limit is close to 10 (this limit is where the algorithm derives its truncated label from). The idea behind the binary backoff algorithm is to spread further and further apart in time the retransmissions of collided stations, using the notion of repeated collisions as an indication that the

network load has increased. Interestingly, this algorithm gives approximately last-in-first-out (LIFO) packet transmission ordering, because it gives more transmission opportunities to newer packets.
The mechanisms described above together constitute the unique Ethernet random access protocol, termed *carrier sense multiple access with collision detection* or CSMA/CD, which sits somewhere in both the physical and data-link layers of the OSI seven layer framework model. Note that because of this trait, the CSMA/CD scheme is independent of the medium, and can be applied to any broadcast channel such as radio, twisted pair, coaxial cable, fibre optics or diffuse infrared. Shoch and Hupp point out that coaxial cable is well suited to constructing a local area network (high bandwidth communications for machines within a distance of approximately 1 kilometre).

The intensive study presented in [SHOC 80] examined in detail the operation of a local Ethernet computer network which had been in use for a number of years. The following important findings were made about the performance of an Ethernet system:

- the error rates are very low and very few packets are lost.

- under normal load, collisions are very rare and latency is virtually zero.

- although there are more collisions under heavy load, the collision sense and resolution mechanisms work efficiently, keeping the broadcast channel utilisation very high - approaching 98 per cent (compared to the very small 18 per cent figure for pure Aloha channels [ABRA 77]).

- even under extreme overload, the Ethernet channel remains stable.

3.3.4 Medium Access Protocols Used in Metropolitan Area Networks (MANs)

MAN network design is quite different to that of HFC, satellite or wireless networks, as has been described in [SALA 95]. Although MAN networks share with the other network types (except wireless) long propagation delays and must be designed with *basic message unit* (an atomic message which can be sent in one transmission time) transmission times to significantly exceed the longest one-way propagation delays, they actually have an advantage over the other network types: the physical medium in MANs may be chosen to achieve maximal efficiency. The architecture of most MANs is based on a unidirectional medium, whether it be a unidirectional bus, dual bus, ring, folded bus, or a buffer insertion ring. Table 3.4 (overleaf) shows a summary of the medium topology and other main properties for some common MAN protocols, the details of which are given in [SALA 95], which is also where the table was reproduced from.

A property of any unidirectional medium is that it implicitly defines an order for the stations, which may cause significant problems in fair sharing of the medium. Intuitively, the stations closer to the slot

generator may reserve all or most of the bandwidth unless a further control mechanism is implemented, an issue we discuss in Section 3.3.4.1 for the MAN protocol which eventually became part of the IEEE 802.6 MAN standard - that is, the Distributed Queue Dual Bus (DQDB) [MAN1 89]. The basis of such a control mechanism is that the stations closer to the slot generator should abstain from transmitting even if their queues are not empty, and firstly confirm that the whole channel is idle before attempting transmission. This then allows the stations further downstream an opportunity to transmit, rather than to always be waiting for upstream stations to finish their transmissions first.

Protocol ⇨ **Characteristic ⇩**	***Fasnet***	***Expressnet***	***Orwell***	***Cyclic Reservation Multiple Access (CRMA) II***
Control	Centralised	Distributed	Distributed	Centralised
Topology	Dual Bus	Unidirectional Bus	Slotted Ring	Buffer Insertion Ring
Transmission	Synchronous	Asynchronous	Synchronous	Synch. / Asynch.
Cycling	Voice ⇨ New Bursts ⇨ Data	Voice ⇨ Data	Mixed Voice and Data	Mixed Voice and Data

Table 3.4: Summary of MAN MAC Protocols

As [SALA 95] points out, an important step towards implementing a fairness mechanism is the creation of a *cycle* concept. Namely, stations can only transmit a pre-determined number of cells per cycle; this is the system's implementation of a QoS mechanism aimed at guaranteeing a minimum amount of bandwidth per station. This cycle concept in the design of MAN protocols is the foundation for issues of framing, frame synchronisation and multi service class support in HFC networks, as discussed later in Section 3.3.5.2.

3.3.4.1 MAN Protocol Standard: The DQDB Protocol

The Distributed Queue Dual Bus (DQDB) protocol was specified by the IEEE 802.6 project team as part of the proposed standard for the interconnection of Local Area Networks (LANs), computer mainframes and other devices, known as the Metropolitan Area Network (MAN) standard [MAN1 89]. In addition, prior to Asynchronous Transfer Mode (ATM) technology coming to the fore, the ANSI T1S1.1 committee working on the User Network Interface standard considered DQDB as a major component in the Broadband ISDN (B-ISDN).

DQDB uses a packet access scheme based on dual point-to-point unidirectional buses which carry traffic in opposite directions past each station. The paper by Zukerman and Potter [ZUKE 90] discusses the performance characteristics of DQDB under overload conditions, and suggests that fairness is an issue in this protocol. Namely, [ZUKE 90] states that there exists a trade-off between efficiency and fairness. Use of the Bandwidth Balancing mechanism eliminates the positional advantage which some stations enjoy due to the design of the DQDB protocol, at the price of sacrificing bandwidth.

The example given in [ZUKE 90] is that of two stations both wanting to transmit at peak channel capacity, with one enjoying an unfair positional advantage over the other. If a fairness mechanism is used (e.g. Bandwidth Balancing) then the stations will share capacity reasonably equally, *but they will both take longer* to complete their transmissions of a large file, due to the loss of bandwidth involved in any such fairness enforcement scheme. This leads to an interesting issue, which is generically applicable to all protocols: *"why give away capacity for the sake of fairness, when you need capacity the most?"*. In the quoted example it is apparent that a fairness mechanism achieves nothing. This is clearly a situation where some kind of compromise needs to be found, between optimising the system globally (i.e. do not waste bandwidth) and optimising it for the users locally (i.e. do not allow one user to selfishly command an unfair share of system bandwidth).

3.3.5 Medium Access Protocols Used in Hybrid Fibre/Coax Networks

3.3.5.1 Background and Architectural Information

Satellite and HFC networks tend to bear the most similarity to each other, due to the almost identical restrictions imposed by the two architectures on the nature of communications between stations. For example, in satellite communication systems it is the lack of line-of-sight between stations which makes communication via a central ground station a necessity; in HFC systems, it is the presence of amplifiers and bridged taps, as shown in Figure 3.11, which makes it all but impossible to communicate directly between stations, and hence necessitates communication to always go through a central Head-End.

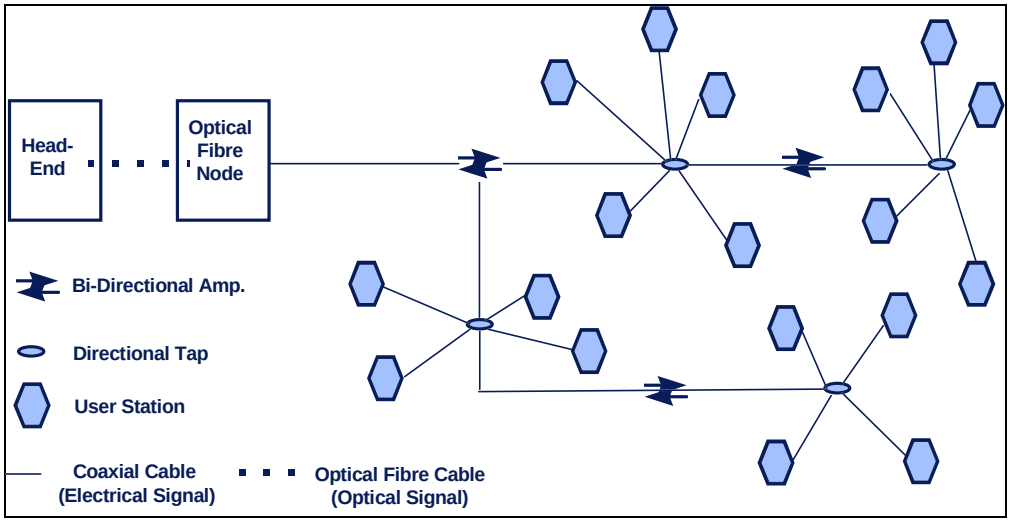


Figure 3.11: A Single HFC Fibre Serving Area - Physical Topology and Components

Wireless networks tend to be similar to their HFC and satellite counterparts, with the exception that propagation delay is usually assumed negligible. HFC, satellite and MAN networks all have significant propagation delays on the other hand, and require the protocols to be designed in such a way that the length of transmission of the *basic message unit* significantly exceeds the longest one-way propagation delay, in order to achieve desirable efficiency. A good example of this is the UniLINK protocol (see Section 3.3.5.3.2), where the minimum message length must not fall below the maximum round trip delay.

As shown in Figure 3.11, the dashed line joining the Head-End to the Fibre Node, is the optical fibre network segment. This part of the network cable carries the signal in optical form. The coaxial network segment carries a converted electronic signal from the Fibre Node outwards into the actual Fibre Serving Area (FSA). Usually there is about 2kms from the Fibre Node to the furthest station (limit of FSA) which has been passed by the coaxial cable. However, it has been suggested that the IEEE 802.14 standard should provide support for this distance to be up to 16kms (as mentioned in Section 3.2.1.4.2).

Clearly in this type of shared medium system, no user has a dedicated line of communication to and from the regional Head-End (or even to the Fibre Node for that matter). The system logically resembles a dual bus architecture, but unlike DQDB for example, the topology is very different, being that of a tree and branch variety. What may not be apparent merely from looking at Figure 3.11, is that the signal may flow easily from the stations to/from the Head-End. However, the presence of the directional taps and amplifiers (which separately act on up/downstream signals) highly attenuates any signals which may attempt to pass *between* individual stations. Thus any communication between stations, even if they are next to each other, has to always travel upstream to the Head-End and then be broadcast by the Head-End back onto the downstream channel.

This feature makes the topology two uni-directional buses, and analogous to the just mentioned wireless/satellite communication systems, where all communication must go through the base or ground station. Due to the read and write restrictions and rules imposed on the two uni-directional buses (e.g. a station can never write downstream, all upstream writes must go to the Head-End, no sensing of upstream channel is possible), the efficient use of existing shared media protocols like DQDB and CSMA/CD is ruled out. In particular, DQDB can not efficiently be used on tree-branch network topologies, such as the one used in HFC networks.

The need for controlling upstream access to the medium arises from the simple fact that if a user could start to transmit their data whenever they wanted, a problem with repeated collisions would arise. It would be especially problematic trying to detect collisions using something like the CSMA/CD approach. Namely, the need for a suitable protocol to support coaxial cable distances in excess of a few kilometres makes the propagation delay an inhibiting factor in the use of the same CSMA/CD protocol as used in Ethernet LANs. The IEEE 802.14 standard provides for the possibility that the "last coaxial distribution mile" to the home can be up to 16 kms of total coaxial cable length. This is certainly in contrast to the 802.3 Ethernet specifications where a single LAN segment (i.e. one collision domain) usually does not exceed 500 metres. With very large propagation delays, even at relatively light loads, there is little chance that an entire data unit (cell or packet) will propagate across the shared medium to its desired destination, before another station attempts to send information.

In addition to an HFC system’s long propagation delay, the Head-End *processing time* must also be accounted for, and so it is important to know what typical Head-End functions tend to incur this time penalty. Firstly, when the Head-End changes from reading the line signals of one station (*listening*) to listening to another station, it has to take some time to *tune in* to the inevitable differences in the incoming

signal. The amplitude of the new signal will be different, and in addition to this, the Head-End must achieve bit-timing, and framing, before it can interpret the incoming digital information. Note that this is not the case for stations, because they are permanently tuned in to the Head-End's unvarying signal in the downstream. Apart from this physical component, the Head-End processing time is also comprised of logical functions such as station address lookup, bandwidth management of both stream-oriented and packet based connections, and synchronisation and scheduling of the stations' upstream transmissions.

3.3.5.2 Generic HFC MAC Protocol Design Issues

A primary goal of an HFC MAC protocol must be to economically exploit the inherent shared infrastructure of this type of network. It should thus be as inexpensive as possible to design the user station, of which there will be many, while the Head-End complexity level can be arbitrarily high, given that the cost of this one unit is amortised by all the stations in a number of FSA clusters. As [LIMB 95] and [SALA 96a] point out, the other main goals of a generic HFC MAC protocol are that it:

- exploits the medium efficiently over a range of transmission speeds (1-100 Mbit/s) and over relatively long distances (up to a maximum combined fibre and coax distance of up to 80km);

- has relatively low latency (defined as the average waiting time to access the medium and successfully start transmitting) in order to support interactive applications and voice - a few tens of milliseconds would be the upper limit;

- supports the ATM constant bit rate (CBR) service class - also known as fixed rate information streams;

- supports multiple priorities for non-CBR service classes.

3.3.5.2.1 Reservation Based Paradigm

A HFC protocol must, by the very nature of the system it operates over, be *reservation based*. A small initial message, usually called a *request*, is sent in order to get system permission for further guaranteed resources. If this was not done, then the two already-mentioned problems of long propagation delay and the impossibility of sensing the medium prior to transmission, would make the frequency of resulting collisions intolerably high. Although there is contention among the request messages, and they can suffer collisions, they are much smaller than the user data messages, and collision free data transmission is assured. However, some protocols do specifically try to cut down the overall latency in waiting for the reservation phase to start, by permitting the initial request to also carry the first (or sometimes, the one and only) data message. This is termed *immediate access mode*, due to the ability to send out cells immediately upon their arrival at a station. It should always be remembered that as in any design process, an eye must be kept on what aspect of performance is being sacrificed in order to improve another. In the case of *IAM*, the cost is the added burden of actually suffering collisions in the much larger data slot, which has obvious ramifications on bandwidth wasted due to retransmission requirements.

The generic slot structure used in most of the MAC protocol proposals presented below, and indeed in the draft standard, is shown in Figure 3.12 (overleaf), as reproduced from [SALA 95]. Note that there are two types of the smaller minislots, which carry the request message. The first of these will from now on be referred to as *Contention Mini Slots (CMS)*; these are only logically part of the larger upstream slot (as shown in Figure 3.12) and still require separate guard-band and header overheads - hence they have been drawn with gaps in between.

The second minislot type is the *Data Mini Slot* (DMS), which enjoys the added efficiency of no guard-band, since it is part of the same station's *unit transmission block* (comprised of Data plus DMS). Since the same station is always responsible for writing a single unit transmission block, no guard-band is required between the Data and DMS regions.

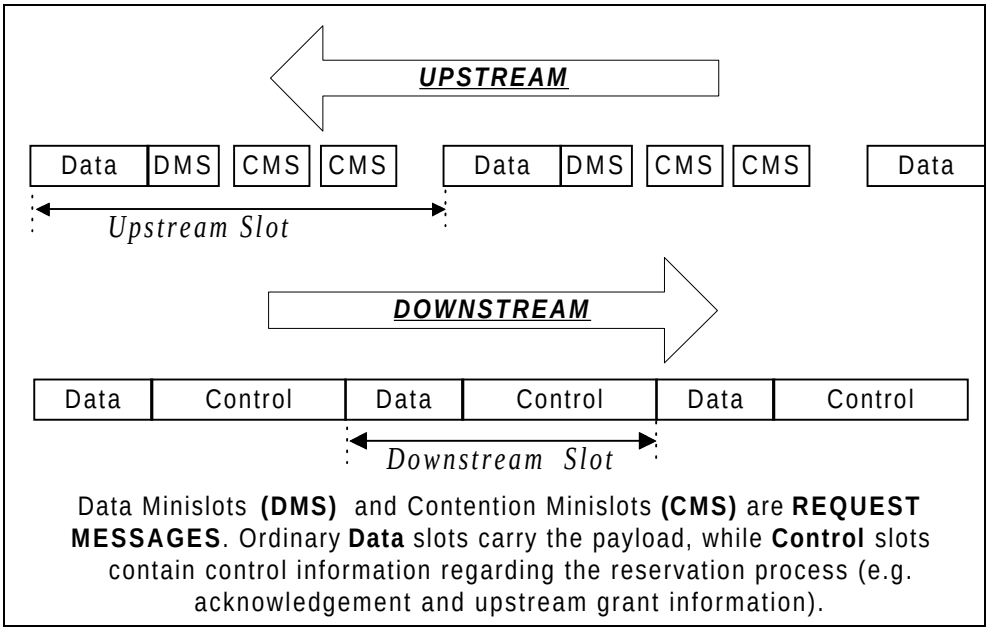


Figure 3.12: Generic HFC MAC Protocol - Slot Structure

The CMS is a different prospect, since any number of stations, with potentially different signal levels and synchronisation, may transmit within it, as it passes by on the upstream propagation channel. This is the reason why the CMS have been drawn in Figure 3.12 with blank gaps between them and other parts of the Upstream slot. As was mentioned earlier, each station is always tuned in to the identical signal emanating from the Head-End, hence no guard-bands whatsoever are drawn between the Downstream slots, which contain the Data and Control slots. As their names suggest, the former carries user data payload units such as one or more ATM cells, while the latter contains system wide information regarding the reservation process.

3.3.5.2.2 Protocol Control - Centralised versus Distributed

The two approaches used in HFC MAC protocols rely on totally different paradigms, but it has generally been claimed in the literature that the cost savings in a centralised architecture outweigh most of the benefits of a distributed one. The cost savings arise, as was mentioned earlier, due to the fact that any

complexity in the protocol which is meant to reside in the Head-End is fully spread cost-wise over all the users of the FSA cluster. Distributed protocols do provide advantages, but because most of the reservation is controlled at the stations, the increased customer premises equipment (CPE) cost does tend to produce a higher budget solution. It is for this reason that virtually all protocols which have been submitted formally to the IEEE 802.14 WG are based on a centralised control.

A **centralised protocol** is one in which the control information is issued to all stations by the Head-End, after it has processed any requests and analysed system state information. The Control slot in Figure 3.12 is used to convey this information back to the stations. As explained in [SALA 95], the two approaches to the generation and delivery of control information, which are adopted within a centralised protocol are *Server-wait* and *Station-wait.* In the Server-wait strategy, the Head-End computes the slot assignment and waits for the station's turn to come up in the upstream schedule before delivering it, taking into account differences in propagation delay if necessary. The Station-wait scheme involves some more complexity at the station - it is delivered its assignment ahead of time, and then must perform its own countdown to the upstream transmission event. This latter scheme only has one advantage, and that is that control slots are shorter because they are sent immediately and to individual stations. The former scheme however, has the distinct advantages of minimal intelligence at the station (it just receives a grant and sends its allocated number of slots upstream beginning immediately), as well as the flexibility of last minute Head-End changes in the slot assignment, when new request arrivals change the system state (especially multi-priority systems).

A **distributed protocol** is one where each station receives in the downstream Control region a copy of the current upstream minislot "snapshot" (of course no guardbands are needed between the minislots in this Control slot, since there is one predefined transmitter signal - that of the Head-End). With this snapshot, each station is made aware of the upstream minislot state, and may make its own independent decision about what to do with any given upstream minislot. Thus the slot assignment process is duplicated and computed at every station, making it a necessity for the system to ensure that all stations arrive at the same slot assignment! A key advantage of this type of control, is that the number of collisions can be minimised with this variety of "pre-emptive minislot occupancy" knowledge.

3.3.5.2.3 Differences in Propagation Delay (DPD) versus Virtual Delay Buffer (VDB)

There are two distinctly differing schools of thought regarding this issue. The first says that all stations are "virtually moved" to the end of the distribution line [SLOS 96], by introducing a position-dependent *Virtual Delay Buffer (VDB)*, whose length (in time) varies from zero for the station furthest from the Head-End, to the full propagation delay for stations nearest to the Head-End. An obvious advantage of the VDB approach is that it lacks scaleability due to its total dependence on the station furthest from the Head-End being the reference point. Thus if the coaxial distribution network keeps growing, as it is expected to do, and the furthest stations grow to be more and more distant, the VDB of each station needs to be constantly re-adjusted upwards. This represents an ongoing increase in the mean access delay of messages, by a value which is related to the buffer size [SALA 95].

The second school of thought centres on tolerating the system's Differences in Propagation Delay (DPD). The main disadvantage of this strategy is that many existing contention resolution algorithms (most of which are efficient and attractive to the protocol designer!) are made specifically for constant propagation delay systems, and cannot be applied to a DPD network without major rework. It is stated in [SALA 95] that in this case, it is a more common design decision to introduce the delay buffer rather than to modify the CRA.

An advantage of a DPD system lies in its ability to emulate a multiserver system, as compared to its VDB counterpart. Namely, in [LIMB 89], it has been shown that a constant propagation delay access network behaves like a G/D/1 queue under low load and arbitrary traffic, or like an M/D/1 queue under Poisson traffic specifically. A system with DPD would under the same low load and arbitrary traffic conditions behave like a G/D/*m* queue, with *m* being the maximum propagation delay difference in slots (under the assumption as stated in [LIMB 89], that there is at least one station per coaxial cable slot along the line).

This means that unlike the single server model of a VDB-adjusted constant delay network, a DPD system may tolerate many simultaneous single-slot transmissions, with suitably long idle periods between these burst arrivals, in order for the slots to propagate to the other end unhindered by other arrivals (without collisions). On the other hand, the constant delay network is more suited to a staggered arrival process, something akin to Poissonian traffic, where the variance to mean ratio is not too high and arrivals are spread over time, because any time that more than one station requests service within the same slot time, a collision occurs. In the queueing models, there is no difference between the G/D/1 and G/D/*m* systems' service rates, because each *can "serve" one station per slot on average*. However, in any bursty traffic environment (an emerging observation of data traffic behaviour in modern computer networks, as was detailed in Section 1) the DPD system will be less prone to collisions, since its bursty method of service better fits such a traffic arrival process.

3.3.5.2.4 Choice of Contention Resolution Algorithm (CRA)

The design issues which we have explored so far in Sections 3.3.5.2.1 to 3.3.5.2.3 all have a direct impact on the collision probability experienced within the access network. But once a collision has already occurred, and we wish to define sensible retransmission rules, we enter the domain of the contention resolution algorithm (CRA). The most popular CRAs were described in Section 3.3.2, and here we discuss HFC-specific matters.

Upon surveying the properties of the most well-known CRAs, as shown previously in Table 3.2, a few things become evident. The potential for CRA instability has led to the concept of the piggybacked DMS being proposed for the future IEEE 802.14 draft standard. In addition, it would appear that the attractively high throughput and inherent stability of the splitting-based CRAs, has prompted the IEEE 802.14 working group to propose this type of CRA for inclusion into a future draft standard. The CRA with most widespread acceptance among the IEEE 802.14 working group at this moment, is a stack-based, free-access splitting algorithm based on Bisdikian's START presented in [BISD 96a], but with the added

option of blocked-access operation (i.e. so that it can operate as a tree-search algorithm as well). The major implications of this decision are that DPD is not supported, while robustness is relatively good, due to the limited sensing nature of START. As has been explained in the previous sections, the need to support DPD is eliminated by the introduction of a VDB, while robustness can be further improved by (i) ensuring minimal errors through various error detection/correction schemes at the physical layer and (ii) implementing a sophisticated centralised "re-negotiate" feature which can re-synchronise the system when it enters or nears deadlock situations. This shows us that except for achievable throughput which is a given for any CRA, most of the negative aspects of any one CRA may be adequately compensated for by the introduction of various "add-on" mechanisms or preventative schemes at other layers.

On a final note about CRAs, the very long propagation delay typical of HFC networks has forced the inclusion of the contention interleave (CI) concept into proposals for the draft standard. The proposed concept is that network-wide it is possible to have many (more than ten) interleaved contention processes active at a given time, although each station must have a separate contention state machine for each interleave it wishes to contend in concurrently. It is anticipated that most networks will have only a few contention interleaves. It is hoped to introduce sufficient scope and flexibility into the draft IEEE 802.14 standard with this proposal, that the manufacturers of equipment may opt for varying levels of complexity - especially at the stations, which would be the mass-market deployed units (and where the support of a large number of CI processes would come at a significant cost).

3.3.5.2.5 Station Addressing Security

In the downstream direction, all information is broadcast from the Head-End, thus no contention resolution is necessary since all stations "snoop" the medium, but only read message units or slots actually addressed to them. Thus, apart from the other design considerations, any HFC protocol has to cope with the important issue of secure station addressing (a station must only read its own data). This is something which could potentially be a problem if user equipment begins to unintentionally or maliciously read the downstream transmissions of other stations, by having its Station ID changed. The feasibility and extent of such a threat is something for further investigation, and not much has been published about it in the literature.

3.3.5.3 Survey of Current HFC MAC Protocols

The IEEE 802.14 working group (WG) *"MAC and PHY for Hybrid Fibre-Coaxial Systems"*, is still in the process of finalising the standard definition. Outside of formal submissions to this IEEE committee, not a lot of detailed work has been published in the area of HFC MAC level protocols and their performance analysis. Some of the few exceptions are DQRAP [XU 93], XDQRAP [WU 94], PDQRAP [LIN 94], CPR ([LIMB 95], [SALA 96a]) and MLAP [BISD 96a]. In addition to these publications, at the 2nd International Community Networking Workshop in 1995, LanCity corporation gave a general presentation of their cable modem product and the UniLINK protocol [ULM 95].

The other major candidate MAC protocols for HFC networks discussed here, are all based on the presentations made in the IEEE 802.14 WG committee standard meetings, with a restricted disclosure of the proposals.

Protocol Properties	PDQRAP	①CPR and ②MLAP	ADAPt+	FPP	SR-GRAP	UniLINK
Control	Distributed	Centralised	Centralised	Centralised	Centralised	Centralised
Reservation Mechanism	Slotted ALOHA	Slotted ALOHA	Slotted ALOHA	Group Polling	Group Polling	(modified) CSMA/CD
Number of CMS	4	① 1 ②>1	>1	1	1	0
Contention Resolution Algorithm	Ternary Tree	① p-persistent ② *n-ary* Stack	Binary Exp. Backoff	Polling Binary Tree	1-persistent	Unspecified
Piggybacked DMS	No	Yes	Yes	Yes	No	No
Immediate Access	Yes	No	Yes	No	No	Yes
Propagation Delay	Fixed	① Variable ②Fixed	Fixed	Fixed	Fixed	Variable
Interleaving	Yes	No	Yes	No	No	No
Slot Size	Fixed	① Fixed ② Variable	Variable	Variable	Fixed	Variable
Frame Size	Not Applicable	Fixed	Fixed	Fixed	Variable	Fixed
Concatenation	No	No	Yes	No	No	Yes
Frame Synchronisation	Yes	① No ② Yes	Yes	Yes	Yes	Yes
Frame Format	clustered	mixed	clustered	clustered	mixed	clustered
Service Classes Supported	CBR^{-} $VBR^{NRT+RT-}$ $ABR^{\sim}$ $UBR^{\sim}$	CBR^{+} $VBR^{NRT+RT-}$ ABR^{+} $UBR^{\sim}$	CBR^{+} $VBR^{NRT+RT-}$ ABR^{+} UBR^{-}	CBR^{+} $VBR^{NRT+RT-}$ ABR^{+} UBR^{+}	CBR^{-} $VBR^{NRT+RT-}$ $ABR^{?}$ $UBR^{?}$	CBR^{+} $VBR^{NRT+RT-}$ ABR^{+} UBR^{-}

Legend: *"+" the protocol can support the service class with no modifications.*
"-" the protocol can support the service class with some modifications.
"~" the protocol can support the service class, but inefficiently.
"?" not enough information is available to decide.

Table 3.5: A Comparison Between the HFC MAC Protocols

[SALA 95] contains a detailed survey and summary of six prominent HFC MAC protocols, as they stood in mid-1995. Table 3.5, reproduced from [SALA 95], shows the characteristic properties of the seven main HFC protocols (MLAP included), as well as their ability to support the five standardised ATM Service Classes [ATF2 95]. Although [SALA 95] was a valuable information source at that time, major changes have since occurred, such as the formal submission and then public release of IBM's MLAP protocol in 1995 and 1996 respectively (it became the seventh major HFC MAC protocol which we consider). In addition, the formerly AT&T and then Lucent Technologies protocol submission, named ADAPt, was extensively revamped (and called ADAPt+) to improve its CRA stability and efficiency in handling both ATM and IP type traffic, with QoS guarantees.

Most of the protocol property definitions in the leftmost column of the table are self-explanatory, so below we just clarify the ones which may not be immediately obvious:

- **Immediate Access -** ability to send a data rather than a bandwidth request message immediately onto the shared medium, thus risking collision of the data message itself rather than just of the signal message.

- **Interleaving -** whether it is possible to concurrently implement several copies of the basic contention resolution algorithm network-wide, so that stations may participate in more than one "*contention state machine*".

- **Concatenation -** protocols with this capability allow a station which has built up more than one message in its queue, to concatenate more than one message into one "super-transmission", thus saving on header overheads and reducing collision probability.

- **Frame Synchronisation -** whether the upstream and downstream frames are time-synchronised.

- **Frame Format -** whether there are specifically allocated frame regions for different traffic types (clustered) or whether the different message types fill the frames on a mixed ad-hoc basis (mixed).

Although large parts of Table 3.5 are reproduced from Table 4 within [SALA 95], we have made some key changes such as the re-evaluation of the newer ADAPt+ protocol, and the consideration of CPR and MLAP as one "type" of MAC protocol. The latter decision was made because, as Section 3.3.5.3.1 will show, apart from the CRAs used and some other non-essential differences, the two protocols bear a very high degree of similarity both to each other and to the December 1996 draft IEEE 802.14 standard. It is also for this reason that we spend more time giving an overview of the Centralised Priority Reservation and MAC Level Access protocols, than we do on some of the others.

3.3.5.3.1 Centralised Priority Reservation (CPR) and MAC Level Access Protocol (MLAP)

The Centralised Priority Reservation (CPR) protocol proposed in [SALA 96a] and IBM's MAC Level Access Protocol (MLAP) proposed in [BISD 96a] are two very similar protocols. One of the major differences is that MLAP has been formally documented and submitted by IBM to the IEEE802.14 WG for consideration (as a candidate for the standard). The similarities between CPR and MLAP are many. Both support multiple data structures within their MAC frames. Both have multiple priorities as well as isochronous stream support in order to support Quality of Service (QoS) constraints for various classes of traffic. In this sense, either protocol may be said to be "ATM-friendly". This explains the fact that both protocols rely on overlaying a frame structure on the component time slots, in order to achieve stream-support. The two protocols also both provide support for multiple outstanding (unacknowledged) messages, but restrict themselves to one outstanding message per priority level (i.e. traffic class).

Also, in general, the ***mode of operation of both protocols is almost the same***:

- Inactive stations, or, those that have nothing to transmit are not allocated any bandwidth, (with the exception of some Constant Bit Rate (CBR) guaranteed bit-rate services).

- When a station receives new traffic, it attempts to signal its bandwidth request to the Head-End, via a random access contention portion of the channel.

- When collisions are successfully resolved and the requests reach the Head-End, a bandwidth controller appropriately allocates and schedules future transmissions from the stations. Note that this apportioned upstream bandwidth is explicitly ***reserved*** for specific stations, in both protocols.

At this point, it is significant to note that although most bandwidth schedulers within the Head-End will be guided by the required QoS constraints, the specific traffic scheduling algorithm used is not part of either of these MAC level protocols - and it will not be included as a part of the IEEE 802.14 draft standard. The algorithms in the Head-End controller are left unspecified as much as is practical. This should ensure interoperability among modems while leaving each vendor free to add value through their own Head-End implementation.

- By means of the downstream channel, specific stations are notified of the time of their allocated transmissions and the allowable quantity of data they may transmit.

- If a station which is still active generates and enqueues a further amount of traffic, during the period that it is transmitting in its own reserved upstream frames, it may immediately request, without resorting to the contention part of the channel, more bandwidth. This “piggybacked” mode of requesting is also specified in the draft standard, and it may go on until the station's queue has been emptied.

The ***differences*** between these two protocols lie in the following areas:

- MLAP has fixed length "blocks", of a few milliseconds duration, within which variable length "slots" for reserved and contention data may be dynamically allocated, according to the traffic conditions. CPR on the other hand has much smaller, fixed length data slots, separated by one or more contention minislots (for the signalling) in the upstream, and by one or more acknowledgement/grant minislots in the downstream path.

- MLAP claims to support flow control (a probability field with four settings from 0% to 100%) for newly arriving traffic.

- In CPR, stations may at any time transmit within the next arriving upstream contention minislot, while in MLAP, stations are explicitly told which slots within a block are available for them to contend within. The draft standard specifies a contention process akin to this latter MLAP process.

- MLAP may support a more than one acknowledgement slot within a single downstream block - for "fast response". However, the size of an MLAP block is much larger than that of a downstream data slot in CPR, hence the number of acknowledgement messages either protocol may send in a given unit of time is probably going to be of the same order of magnitude.

- CPR uses a Slotted Aloha -type random access mechanism, with p-persistence (retransmit in the first slot after collision is confirmed, with probability *p*). MLAP, on the other hand, uses the *START-n mechanism (n-ary STAck ResoluTion)*, which has been shown by work referenced within [BISD 96a] to have a slightly higher maximum attainable throughput that that of stabilised p-persistence Slotted Aloha (about 0.40 packets/slot as opposed to about 0.37 packets/slot). It is a separate issue as to whether this relatively small throughput gain when in contention mode is worth the added complexity of START-n, especially given that once a station has control of an upstream transmission path, it may always clear its queue. The START-n algorithm can switch between blocked- and free-access modes depending on the traffic demand, and can also operate over a dynamically changing number of contention slots per block. As mentioned earlier, a hybrid blocked- / free-access splitting algorithm similar to START-n is currently the strongest contender for the draft IEEE 802.14 standard.

- The MLAP specification submitted to the IEEE 802.14 WG contains full provisioning for error recovery (timeout, lost frames, corrupted frames etc.). This is something which is necessary for a full submission to a standard-defining body.

Note from Table 3.5 above, that while the Stack and Tree -type CRAs are both stable, the fact that there is another load control mechanism present in most HFC protocols renders this stability not strictly necessary. For example, the “piggybacked” method of reservation is a stabilising load controller which renders contention under overload unnecessary. Unlike the MLAP protocol and that described in the standard, CPR uses the inherently unstable but very robust and simple p-persistence CRA, thus capitalising on the stabilising property of its piggyback reservation process.

3.3.5.3.2 UniLINK

UniLINK [ULM 95] is an interesting centralised protocol, in that it functions by suitably modifying the Ethernet CSMA/CD algorithm detailed in Section 3.3.1, so that it can function efficiently in an environment with long propagation delay. The protocol senses the downstream channel, and using knowledge about its own round trip delay, RTD, and system maximum round trip delay, RTD_{max}, as well as the length of the passing message, it determines when it is safe to transmit upstream, collision-free. Collisions are still possible because more than one station may decide to begin upstream transmission simultaneously. Note than the “vulnerability period” is RTD_{max} [ULM 95], and all protocol messages (downstream and upstream) are longer than RTD_{max}, with the length included in their header.

Another unique facet of UniLINK is that the central node, called pacer, is not necessarily located at the Head-End, meaning that the Head-End is just like any other node (except that it houses the scheduler and

bandwidth manager). The pacer is responsible for generating the *Block Synchronisation Packet (BCP)*, which identifies the beginning of a frame, and is used by other stations for estimating their own RTD values. The *BCP* also includes the assignment of slots within the next upstream frame. In addition to a *BCP*, there are three other distinct frame regions: Contention, Periodic Dedicated and Reservation Dedicated. Within the Contention region, we find non-delay-sensitive traffic and bandwidth reservation requests, following the above mentioned modified CSMA/CD algorithm. The Periodic Dedicated region is used for delay-sensitive traffic and once a station obtains a reserved portion of this region, it keeps it for the call duration. Finally, the Reservation Dedicated region reserves bandwidth on a burst on-demand basis, rather than for the entire connection duration.

The UniLINK frame length is constant, but it contains variable length messages, in order to allow concatenation transmissions. That is, several messages currently in a station's queue may be sent out as one long concatenated transmission (each message contains its destination address to facilitate receiver decoding), achieving increased efficiency both due to (i) a reduction in overhead headers and (ii) a reduction in collision probability due to increased total transmission size.

3.3.5.3.3 Spatial-Group Randomly Addressed Polling - SR-GRAP

The basic premises SR-GRAP [CHEN 95b], a centralised protocol, is built on, are the usage of polling, orthogonal line codes (to decode simultaneous transmissions) and a three level hierarchical grouping of stations based on propagation delay. The latter aspect allows this protocol to use to its advantage the DPD between stations scattered along the coaxial distribution network. The primary level of the hierarchy defines *mega-groups*, which contain all the stations with identical propagation delay (i.e. no discernible DPD between them). Thus, when the Head-End polls stations within different *mega-groups* there is no chance of a collision even if the stations reply immediately. The *mega-groups* contain *super-groups*, each of which has stations tuned to transmit at a given power level. Stations from different *super-groups* may transmit simultaneously without causing a collision. The number of orthogonal codes available determines the number *p*, which is how many unique station addresses in each *super-group* are available for random selection at the time when the stations respond to the Head-End's initial "READY" message at the beginning of their *super-group*'s contention period.

The operation of the randomly addressed polling (RAP) protocol is described in [CHEN 95b]. Prioritised types of traffic (such as voice) get a periodic poll after the call establishment, when the required bandwidth is reserved according to set-up specifications. Note that the protocol has a variable length frame structure with *p* alternating broadcast-polling regions followed by a reservation region. Both the polling regions after the broadcast and the reservation region vary in length - the former because of the number of decoded response addresses, the latter to cater for delay sensitivity of circuit emulation traffic. No piggybacked reservation scheme is used, and the CRA, being unstable (random address polling with 1-persistence), is thus not backed up by a secondary load control mechanism.

3.3.5.3.4 Framed Pipeline Polling - FPP

In systems with small round trip delay (RTD) and always-active stations, polling has been shown to be highly efficient. Since this is not the case in HFC networks, FPP [MOMO 95] is a protocol which overcomes the delay and inactivity problems by utilising a traditional polling mechanism together with a pipeline and framing strategy, which make the protocol effectively a reservation scheme implemented via polling in the CMS. The protocol is centralised, provides for one CMS (also termed *retry poll*) and one DMS, allows variable length messages and cannot support immediate access (as stands to reason in a polling-based environment).

The pipeline strategy revolves around eliminating the inherent inter-poll idle time in a standard polling mechanism (which is a particular problem for large RTD systems), by clustering all polls to be sent in a frame, in one continuous transmission. In this way, the Head-End sends at the end of the current downstream frame all of the next upstream frame's polls, with the station transmission order following exactly the polling order. Thus, unlike in traditional polling, the stations, once polled, wait for their allotted turns to start transmitting, rather than doing so immediately upon receipt of the poll. The problem of bandwidth wastage occurring if a station is not active when it is polled, is rectified by polling ONLY the active stations individually (by station ID) and then polling all the inactive stations at once within the retry poll contention minislot. The already active stations declare they are still active implicitly, by transmitting new data in response to a poll.

More details on the operation of the protocol may be found in [MOMO 95], but it is important to note that a frame length of 6ms is defined in order to cater for multiple traffic types. Each frame has a CBR, VBR-RT, VBR-NRT, ABR and UBR region, which is followed by individual polling messages for the next frame and then by the *retry poll* for inactive stations. The first two traffic types, having the most stringent QoS requirements, are polled every frame, while the next two (VBR-NRT and ABR) are polled according to their minimum bandwidth requirements. UBR is polled on a variable frame frequency, given that each UBR data burst needs to request bandwidth using the *retry poll*.

3.3.5.3.5 Adaptive Digital Access Protocol - ADAPt and ADAPt+

This centralised, variable frame- and slot-size protocol was first submitted by Sriram et al. [SRIR 95] as an IEEE 802.14 standard contribution in 1995, and was then followed by the enhanced version, ADAPt+ [DOSH 96] in 1996. The upstream frame structure provides for two types of region - synchronous transfer region (STR) and asynchronous transfer region (ATR), all packaged in a 2ms upstream frame. The boundary between the region dynamically adjusts according to traffic demands.

The downstream frames are also 2ms and are thus synchronised with their upstream counterparts, but are obtained from 16 subframes of 125μs each. These downstream subframes have a further subdivision within the ATR, into an ATM subregion and a Variable Length (VL) subregion. This separation permits a given cable modem to carry one traffic type or the other or both, in addition to synchronous circuit-

emulation type services carried in the STR. Also, this ATM - VL separation mitigates the need for MAC level support of ATM QoS requirements for a VL data unit (which is likely to be an encapsulated Internet Protocol IP packet). However, one of the additions of the newest version of the protocol is to provide some QoS guarantees for the VL subregion too, thus catering for the resource reservation and QoS requirements of the new IP version 6 network layer protocol.

In the upstream frame, the STR has 27 byte basic units, each of which represents a standard DS0 64kbit/s channel. The ATR has smaller, 12 byte basic units, and combinations of these are used to allocate slots which contain ATM cells or VL payload data units (PDUs). The old protocol (ADAPt without the +) did not have mini-slot based granularity in the ATR. This concatenation in both the STR and ATR, which restricts carried traffic to given multiples of basic units, allows for much simpler slot synchronisation, and avoids headers and guardbands inside the frame.

Within the ATR region there is a variable sized *contention region* and also a variable sized *reservation region*. In the older version (ADAPt) there was no notion of mini-slots for bandwidth requests (i.e. no CMS) and the contention region allowed stations immediate contention access (i.e. actual data would be sent rather than just a future bandwidth request). In the newer ADAPt+ however, mini-slots are used both for bandwidth requests as CMS units, and as a basic unit of bandwidth allocation in the ATR, as explained in the previous paragraph.

The older version of the protocol, lacking both CMS and piggybacked DMS functionality, was prone to "lockout problems" which we now describe. Competing stations would first send a data cell which is subject to collision, and by means of a *continuation bit* would get a reserved slot within the reserved region, in future frame(s). In some cases, this could cause problems. Namely, once a user got through their first cell, it would be possible to keep reserved bandwidth for the duration of the call or data burst. If enough users did this, the system would be prone to saturation, forcing all the free slots within a frame to become used for existing reserved mode stations, and thus making future contention for newcomers an impossibility.

The newest version of the protocol overcomes this problem in the following ways: (i) the concept of the CMS is introduced; (ii) the idea of *continuation bit* is superseded (replaced) by the idea of piggybacked bandwidth request (the DMS). (iii) The Head-End always provides some number of contention request mini-slots in each frame so that new connections can get their requests in. Thus, system saturation is not possible because the Head-End can deny bandwidth requests for existing connections and give bandwidth to newly active connections.

One of the quoted problems of ADAPt (and ADAPt+) [SALA 95] is that the upstream frame structure, particularly the contention and reservation areas within the ATR region, make it difficult to efficiently support UBR traffic. While the Head-End will know which slots are not currently reserved, it cannot predict which slots are now unused but will be needed by CBR/VBR/ABR type traffic in the near or far future. This means that the UBR traffic would always affect the other traffic types. However, the Head-

End bandwidth management algorithms are beyond the scope of the IEEE 802.14 MAC protocol (this was explicitly stated in the draft standard) as they reside above MAC in the OSI seven layer protocol stack, and therefore it may be possible to rectify this problem with a suitable "ATM focused" Head-End bandwidth management scheme. The other point to note is that this perceived problem with UBR traffic is not a major disadvantage of the protocol, given that most of the other protocols we have surveyed (as in Table 3.5) either cannot support it, cannot do so efficiently or cannot do so without major modifications.

3.3.5.3.6 Prioritised Distributed Queueing Random Access Method - PDQRAP

Being the only distributed protocol proposed so far, PDQRAP and its variants [XU 93, WU 94, LIN 94] is such that each station knows when a slot is not reserved, so that immediate access (i.e. sending of data) to the shared medium is possible. However, in situations like this both a data slot and request slot are sent, to cater for the possibility of a data slot collision. The protocol's complexity comes in the form of having to cancel the request, if no collision had occurred. One of the major advantages of PDQRAP is its support for prioritisation of traffic even during the contention process. On the other hand, it does not support piggybacked data minislots. Instead, four contention minislots are used, with the first intended solely for high priority traffic, regardless of the system state. CMS two to four are used for low priority traffic, except in the case of high priority traffic contention occurring in CMS one (when they are used for the contention resolution of this high priority traffic).

3.3.5.4 Supporting Different QoS: Priority and Scheduling Mechanisms

It is extremely difficult to separate the two mechanisms of priority and scheduling, but one thing is clear - they interact closely together in order to provide support for multiple qualities of the service(s) offered to users of an HFC network.

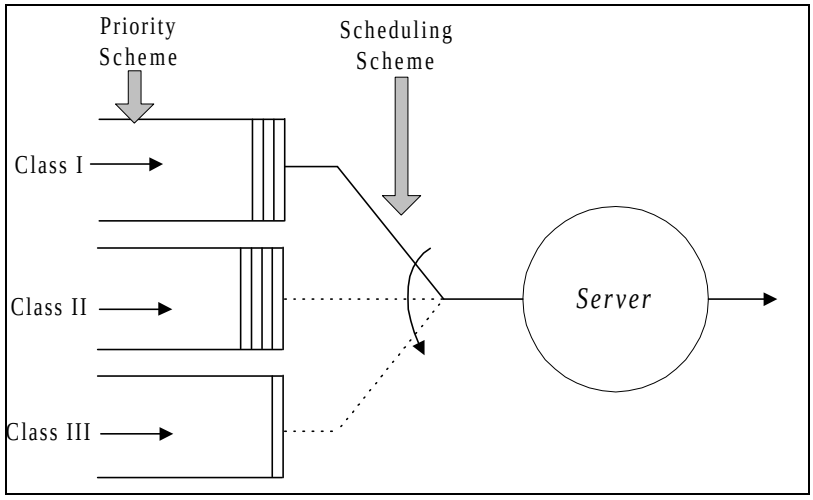


Figure 3.13: Relationship Between Priority and Scheduling Schemes

Figure 3.13, reproduced from [SALA 95], illustrates the relationship: while the *Priority Scheme* resides between the traffic source and the queueing system (and thus defines the rules of the queueing discipline),

the Scheduling Scheme is responsible for the order in which the enqueued message units are served. It stands to reason that the former mechanism must be implemented partly within the MAC protocol itself (via the design of the frame structure), while the latter mechanism of scheduling has to be above the MAC level and implemented at the Head-End processor. The sections below explore some of the important issues related to each of these two types of mechanism.

3.3.5.4.1 The Priority Scheme

The benefit of implementing a MAC protocol with priority is only tangible from the end-user's perspective; the network operator has no direct tangible benefits from such a protocol. This is due to the two differing *system views* that a network operator and end-user have. When looking at the end-user's system view, all we see is a consideration of the specific QoS of his/her end-to-end connection(s) which they are paying for. However, the network operator's system view entails the functioning of the whole network - the QoS requirements of **all** users being currently "served" must be met simultaneously while maintaining fairness between users, while also ensuring that the network operates on a globally efficient level.

By introducing a priority scheme to a basic MAC protocol, the requirements of an individual can be met in a more specific, relevant and granular fashion. The granularity refers to, for example, concurrently offering low, medium and high delay services over a common network infrastructure and thus being able to more naturally seek out and separately target the various "end-user pools" which are differentiated by their QoS requirements and/or their purchasing capacity for the services. This is a particularly tangible improvement on the user side (i.e. a monetary value may be assigned to the benefit of such a service over one with common QoS).

However, the inclusion of a priority scheme does not improve the performance of the existing system, since it is merely either a dynamic or static reshuffling of network resources, such as switching and capacity. The *Conservation Law* explicitly stated in [KLEI 76], points out that rearranging a way that a resource (network capacity) is shared, cannot and does not give a better global service - unless additional resources are added. Another interesting consequence of the conservation law is that any effort to decrease the average delay of a particular user class will simply force an increase in the average delay of some or all of the other user classes. In all of this, the **overall** average delay remains invariant. By completely eliminating a part of a traffic class's delay probability distribution function (pdf) tail, we obtain a certain reduction in average delay and delay variance at the expense of much more significant increases in the delay pdf tails for the lower priority traffic classes.

Although the abovementioned problem shifting excess delays and delay variances from one traffic type to another are the major concern of a priority scheme designer, [VASS 92] outlines some other important issues regarding the design of a priority mechanism and its incorporation into an existing MAC protocol:

- The significant robustness advantage of a system with an LS CRA as part of the MAC protocol, is compromised by the introduction of any priority scheme which heavily relies on a particular frame structure, within which high and low priority traffic regions may need to be separated. This compromise is due to the need for embedded synchronisation flags which aid the stations in detecting such a frame structure. The synchronisation requirement erodes one of the chief advantages of LS schemes over FS schemes: their ability to maintain synchronisation without resorting to Head-End initiated sending of any synchronisation information whether it be in the form of flags or global state update data.

- Hierarchical frame structures such as those of the superframe and frame in ADAPt+ [DOSH 96] where multiple levels of synchronisation are required, may be used in order to lower the delay bounds on some or all traffic classes. This most often comes at the price of an increased overhead, and thus a reduced protocol framing efficiency.

- Often a priority scheme will rely on dynamically adjustable intra-frame boundaries, which divide the various traffic class regions. The cost of this "dynamicity" should not be neglected, because it involves (i) significant computational resources in gathering statistics to arrive at a better-forecast frame decomposition; and, (ii) more capacity needed to transmit these statistics from the stations across the channel to this intelligent station/node (e.g. Head-End) which decides about the next frame-cycle's intra-frame structure.

3.3.5.4.2 The Scheduling Scheme

As was discussed in the previous section, the application of different scheduling schemes, as long as they are based on non-preemptive and work-conserving disciplines, cannot alter the global average delay. They can however alter the delay pdfs, and hence the average and higher order delay statistics, of individual traffic classes. The following is a summary of some major scheduling disciplines as summarised in [SALA 95, PANC 95]. Note that many more than this are possible, and it is often a composite scheme best suited to the environment (which depends on factors such as the priority scheme, customer demand distribution and overall system size) which will best perform the job and achieve the smallest delay distributions.

First-In-First-Out (FIFO) **-** involves the transmission of cells in the exact order in which they arrive. This is a very simple to implement discipline, and it involves simple FIFO queueing, at the price of not being able to support *heterogeneous* environments (those with a population divided into different user classes). However, in homogenous environments (all of the system users are of an identical class), the FIFO scheduling scheme is the one employed most often due to the stated simplicity, and the fact that it achieves the most desirable delay pdf (and hence minimises both average delay and higher order statistics like delay variance).

Static Priority (SP) **-** is akin to a multi queue version of the FIFO scheme, inasmuch that each pre-defined (*static*) priority class is assigned its own permanent FIFO queue holding the arriving cells. The service

discipline is straightforward - the lower priority queues may only start to be served once all higher priority queues are empty. SP is a scheme optimised for tight (low) delay constraint environments.

Round Robin (RR) **-** is a scheduling algorithm relying on the concept of traffic *flows*, which may either be defined as isochronous streams (e.g. connections), or as service classes with several users (the number of which depends on the precision of the QoS attributes). RR then distributes the service time equally between all flows in a round robin fashion. It is said to be "adequate" (meaning non-optimal) when maximum delay (or equivalently minimum bandwidth) requirements need to be met per each flow. Often systems will require the minimisation of *new user interference* (defined as degradation of QoS for existing users due to incoming users). RR can minimise this interference by *isolating* the existing and new user flows. Although RR is a simple to implement scheme for a small number of flows, the complexity rapidly increases with more flows needing to be supported.

Earliest Due Date (EDD) **-** introduces the concept of the *deadline*, being the sum of the arrival time and delay bound. The deadlines are then served in a logical progression (earliest due through to latest due) so that resources are not wasted on serving a request now, given that it has until a lot later to be sent. The EDD scheme is most often used in systems where it is desired to minimise the number of cells which do not meet a target delay bound; furthermore, EDD is recommended when each connection has different QoS requirements. It is intuitive that in homogenous environments, EDD is equivalent to a FIFO scheme.

It is apparent in all of the scheduling schemes we have considered that the chief trade-off is that between simplicity of the scheme and the number of the supportable QoS levels (i.e. the granularity). Although one would naturally expect that a compromise in this trade-off would exist, and be a different compromise for each different protocol, it has been stated in the literature [SALA 95] that *"in order to support highly correlated VBR traffic and bursty traffic, an information based scheduling discipline (closer to EDD) may be required"*. The logic of this statement is that simple schemes such as SP will often give better-than-required QoS to higher priority traffic, at the expense of not being able to carry as much lower priority traffic. On the other hand, information-based disciplines will make better informed decisions about how much bandwidth is required for each priority level so that its QoS is "just right", and hence utilise the network more efficiently, especially when presented with multi-priority correlated, bursty VBR traffic.

3.3.6 Wireless Medium Access Protocols

[BERT 92] describes packet radio networks as multiaccess networks in which not all nodes can hear the transmissions of all other nodes. Furthermore, in such networks, a node's subset of accessible transmitters (those it can hear) and reachable receivers (those to which it can transmit) are in most cases different, due to the varying noise conditions at each of the nodes. The fact that each receiver hears a subset of transmitters rather than all of them, makes packet radio MAC protocols far more complex than the other protocols we have discussed thus far. The reader is referred to [BERT 92] and references therein for more details on generic packet radio networks. However, as stated in Section 3.2.1.5, our main interest lies in

the architecture and associated MAC protocol for what is in our view today's most promising wireless radio technology, WATM. The blueprint for the WATM-based personal communication network proposed in [RAYC 92] and elaborated on in [XIE 95], assumes a micro- and pico-cellular environment in which stations *exclusively communicate to the base station of the cell in which they are presently in*. This architecture model is akin to today's mobile telephony cellular networks, and relies on functionality such as handoff and mobility management, as described in [COX 92]. From the point of view of the MAC protocol, such an architecture simplifies the rather complex notion of receiver and transmitter subsets, and likens the system to the HFC architectures described previously.

One of the current WATM research activities is project Magic WAND (Wireless ATM Network Demonstrator) [WAND 97]. An important system design issue for WAND, and WATM in general is the design of an efficient MAC. Similar to what we have seen in the case of F-CPR implemented over an HFC network, the MAC protocol for the radio interface of WAND is also based on reservation and contention principles. It is called the Mobile Access Scheme Based on Contention and Resevation for ATM (MASCARA) [PASS 97].

In general, other than the physical transmission medium being different for HFC and WATM access networks, sufficiently many network operation principles are the same, that it is not surprising to find many similarities between the family of MAC protocols proposed for WATM systems ([RAYC 92], [PASS 97]) and protocols such as F-CPR or MLAP, which we have discussed previously for HFC networks. The class of MAC protocols in a wireless system primarily depends on the physical layer techniques employed - in the broadest sense, whether we are dealing with a *spread-spectrum*, or, a *narrow-band* system.

If the former of the two methods of modulation is used, Code Division Multiple Access (CDMA) is the de-facto mode of operation. The benefit of using a CDMA scheme is that it can be operated in a "resource shared" packet mode that is quite efficient for the multimedia services scenario: each terminal transmits, using a suitably selected CDMA code, without any coordination with other stations, whenever it has data to send. The significant downside to CDMA systems is the very low maximum bit-rate limit (<1 Mbit/s).

This brings us to the narrow-band modulation method, in which much higher bit-rates may be used (NEC's WATMnet prototype has a radio physical layer bit-rate of around 25 Mbit/s, [NARA 97]), at the price of utilisation efficiency due to the need to employ some method of coordinated shared-medium access. Recall that unlike in spread-spectrum systems, narrow-band modulation assumes that the stations are competing for the same frequency. If only isochronous CBR-type traffic such as voice was to be supported, a static TDMA approach would suffice, with the familiar cyclic reservation of timeslots for each station [BERT 92]. However, the desire to carry multiservice (VBR, ABR and UBR) traffic rather than just voice or just data, imposes the need for some form of dynamic resource allocation, which may be thought of as *dynamic TDMA*.

One of the most significant early dynamic TDMA proposals for the integration of packet data and voice in the wireless scenario, was packet reservation multiple access (PRMA) [NAND 91]. Using some concepts from PRMA, the authors of [RAYC 92] propose a WATM MAC protocol called "Multiservices Dynamic Reservation - TDMA" or MDR-TDMA. The frame format and operation of this protocol is very similar to HFC MAC protocols like MLAP or ADAPt+, whereby the frame is subdivided into request and message slots. Each message slot transmits an ATM-like cell with a data payload of 48 bytes (or a submultiple of 48 / *n* bytes), and associated wireless-specific protocol headers. There is a limit to the number of message slots in each frame which may be assigned for CBR voice traffic, with the rest being open to frame-by-frame assignment to VBR and packet data (ABR, UBR) services. On the other hand, request slots are much shorter in length than their message slot counterparts. They are used, like in most of the HFC MAC protocols we have studied, for initial access in Slotted Aloha contention mode. After a station has initiated contact with the central controller at the BS, it may under certain conditions, use the piggybacked method of reserving future capacity.

A final note about radio access protocol layers: research into the performance of NEC's WATMnet prototype [NARA 97], has shown that due to the highly error-prone nature of the radio medium, a wireless MAC protocol needs to be supplemented with an additional Data Link Control (DLC) layer to ensure adequate performance. The authors of [NARA 97] report that the addition of the DLC layer significantly improves the performance of TCP applications (both delay and throughput), because the DLC layer allows for cell level error recovery. And, cell level error recovery means retransmission of just lost cells, rather than whole TCP segments, which would be the case if no DLC was present and errors were recovered by the TCP layer.

4. Performance Analysis of Slotted Aloha under Extreme Traffic Conditions

This chapter provides the model definitions and derives the necessary fundamental results which will be subsequently used in Chapter 6, in the performance evaluation of deadlock conditions for HFC and WATM signalling channels.

In our analysis of the classical Slotted Aloha multiaccess approach in Chapter 3, it was stated that the assumption of arrivals modelled by a Poisson process, is wholly unrealistic for real traffic and the main reason it was used was to enable analytical tractability. Likewise, we mentioned in Chapter 3 that real systems will definitely be prone to errors due to noise, making the perfect reception assumption seem to be very idealised. The aim of this chapter is to devise a more realistic Slotted Aloha-based model, both in terms of capturing the possibility of channel errors and looking at extremely correlated traffic. In particular, we are concerned with modelling worst-case traffic conditions of an HFC or wireless system's signalling channel (which employs Slotted Aloha with p-persistence CRA), that lead to deadlock. Therefore, we will often refer to the ensuing models as *deadlock models*. Channel errors aside, the principal difference between this deadlock model and classical Slotted Aloha presented in Chapter 3, is that we do not assume Poisson arrivals. Table 4.1 summarises the commonality between the two models.

Classical Slotted Aloha Assumption	***Applicability to Deadlock Model***
Slotted System	✓
Poisson Arrivals	✗ (Instead we use simultaneous arrivals).
Collision or Perfect Reception	✗ (Instead, we allow for channel errors).
Immediate Feedback	✓
Retransmission of Collisions	✓
No Buffering / Infinite Set of Nodes	✓ and ✓

Table 4.1: Similarities Between Classical Slotted Aloha and Deadlock Model

Instead of using Poisson arrivals, we model extreme inter-station correlation through arrivals of capacity request batches. The size of these batches represents a subset of the total station population, as explained in Section 4.1. In all the deadlock models proposed except one, the station population is finite and so the notoriously unstable nature of the Slotted Aloha family of multiaccess algorithms is not a factor by itself (since Slotted Aloha is theoretically stable for a finite population). However, the concepts of theoretical and practical stability are worlds apart, and this paradigm forms a major part of our investigation into the four deadlock models we study. Interestingly, the concept of practical stability also enables us to study an infinite population model, which would otherwise be simply discarded as always theoretically unstable for any background traffic arrival rate. One of the motivations for the development of these models, is what has been coined as the "disaster scenario" within deliberations of the IEEE 802.14 committee [SALA 96c], [BISD 96c], where it is assumed that the totality of stations (say, 200) are all simultaneously powered-up (for example after a power failure) and transmit a request signal in the same slot.

In particular, our Basic Deadlock model is an enhancement of the *fixed p-persistence* CRA investigated in [SALA 96c], which includes signalling channel errors. However, the focus of our models is different: we are not targeting solely the initial station registration process after a simultaneous power-up event, but also the operation of the normal signalling channel under "extreme stress", manifested by inter-station correlated traffic conditions. With this in mind, we propose in Sections 4.2 through to 4.4 three new deadlock models based on the original, that go on to extend our analysis to cases where, aside from the arrivals of large capacity request batches, the Slotted Aloha signalling channel also experiences so-called *background traffic* noise. This type of model has physical significance because it can capture scenarios where one part of the access network stations are still operating normally (and hence provide this background traffic), while all other stations have failed, powered-up, re-registered, and are simultaneously wanting first-time access to the signalling channel. This has been known to happen in some neighbourhoods where power outages can take down only one part of the network - when that part comes back online, a scenario such as the one we have described may occur.

[SALA 96c] gives a particularly simple *adjusted p-persistence* algorithm for the Basic Deadlock model (where the p value is optimised in the central controller and fed back to all stations, on a slot by slot basis), and shows by simulation that the algorithm actually outperforms a deterministic Tree algorithm (based on stations' MAC addresses) in an IEEE 802.14 HFC access network setting. In addition, we recall that Rivest's Pseudo-Bayesian algorithm is another relatively simple way of stabilising infinite-station Slotted Aloha and at the same time maintaining an optimal p value, based on an estimator of the number of currently backlogged stations. The problem is that this technique is derived for Poisson arrivals. Therefore, although estimation-based techniques for the optimal dynamic adjustment of p do exist, they are only applicable under certain assumptions about the traffic. Since these assumptions do not hold for our deadlock models, a dynamic p adjustment algorithm for our models remains for further study.

Note that the particular attention we pay to modelling and exploring deadlock is warranted because, as described in Chapter 3, the piggybacked reservation feature of most HFC and wireless MAC's will ensure relatively "uneventful" and efficient operation of both the signalling and data channel under most normal conditions. The bottleneck, and point of interest, then becomes the set of scenarios where it is impossible to use this piggybacked reservation feature and the entire station population burdens the signalling channel directly (i.e. the deadlock scenario). In the following sections, signalling minislots are referred to as Contention Minislots (CMS's), in line with the terminology of the HFC MAC protocol called F-CPR, briefly discussed in Chapter 3. F-CPR will be our main topic of investigation in later chapters, so this approach ensures a simplified and consistent nomenclature.

4.1 Basic Deadlock Model - No Background Traffic

Let N be the number of active stations. To create the worst-case scenario from a signalling collision point of view, we assume that the N stations request capacity, each for a single-cell message, all at the same time. In other words, they all attempt to write into one CMS simultaneously. From the point of view of the signalling, it seems like a batch of N requests arrives at once for contention resolution. We henceforth use

the concept of *batch size* to describe the number of simultaneously arriving requests. The aim is to derive the mean of the CRI duration, measured in the number of CMS's, ϕ. This mean is designated as $E[\phi] = T_C(Scheme, N, p, P_{err})$, *or* T_C for short, where *Scheme* identifies the signalling capacity allocation scheme used on the signalling channel (see Section 4.5), and P_{err} quantifies the probability of a CMS minislot being errored.

As per Table 4.1, we assume that time is divided into fixed-length intervals (the slotted system assumption). We shall now justify this assumption in the context of an HFC or WATM MAC protocol. Let each of the fixed-length intervals represent one CMS minislot. In practice, depending on the number of CMS minislots associated with each upstream data slot, there may or may not be variable-length gaps between consecutive CMS minislots. This means that by using the notion of the number of CMS minislots elapsed, we are measuring time non-linearly. However, due to the cyclic nature (with a small cycle) of this non-linear relationship in a practical MAC protocol, we can assume linearity. In other words, if T_C is measured by a large number of CMS's, and if the CMS cycle is short then T_C times the mean inter CMS time will be very close to the CRI measured in linear time units (e.g. seconds).

Any given CMS has the possibility of suffering a random error, with probability P_{err}. Whether one or multiple errors hit a given CMS minislot, the effect will be the same (the Head-End will effectively see a garbled collision-like CMS minislot), and so only single errors need to be considered. Since we assume that during each CMS there is an independent, constant probability of an error, the error process is one with geometrically distributed inter-occurrence times (parameter P_{err}).

The variable t represents the number of elapsed CMS minislots from the arrival of the batch of size N. At time t=0, a batch of size N arrives. Let $P(j, t)$ be the probability of having j contending requests at the end of time interval t, (t = 1, 2, ...) and j=1, 2, …, N. We also define $P(0, t)$ as the probability of having no contending requests at time t, for the first time. In other words, $P(0, t)$ is the probability that $T_C = t$. Since all N stations try to access a CMS during the first time interval, and they all collide, we shall still have N outstanding requests at the end of the first interval. Hence,

$$P(N, 1) = 1; \tag{4.1}$$

and

$$P(j, 1) = 0 \quad for\ all \quad j = 0, 1, 2, \ldots, N\text{-}1. \tag{4.2}$$

After the first time interval, all N stations try to access with probability p, but there is also the potential for an error to happen in any given timeslot, with probability P_{err}. The probability of a successful transmission is therefore given by $(1-P_{err}) \cdot Np(1-p)^{N-1}$. In general, when we have j stations contending for transmission, the probability of successful transmission (i.e. a reduction of one in the number of requests "waiting" for resolution) will be given by

$$P_S(j) = (1-P_{err}) \cdot jp(1-p)^{j-1} \tag{4.3}$$

On the other hand, the failure outcome when j stations are contending is defined as the event when the number of outstanding requests does not decrease by one. It occurs either when a single existing request retries and is hit by error, or, when either more than one or zero existing requests attempt to seize the CMS minislot - regardless of whether an error hits or not. The probability of failure is therefore given by

$$\begin{aligned} P_F(j) &= (1 - jp(1-p)^{j-1}) + P_{err} \cdot jp(1-p)^{j-1} \\ &= 1 - P_s(j) \end{aligned} \tag{4.4}$$

The case $j=N$ is an upper bound on the system occupancy and hence a special boundary condition exists: $P(N, 2)= P_F(N)$, $P(N, 3)= P_F(N)^2$, $P(N, 4)= P_F(N)^3$, and in general,

$$P(N, t) = P_F(N)^{t-1}. \tag{4.5}$$

The general case arises when $j= 1,2, \ldots, N-1$, and the state j could have been entered from a higher state or from the same state,

$$P(j,t) = P(j+1,t-1) \cdot P_S(j+1) + P(j,t-1) \cdot P_F(j) \tag{4.6}$$

Finally, the probability of zero outstanding requests at time t is given by

$$P(0,t) = p \cdot P(1,t-1) \cdot (1 - P_{err}) . \tag{4.7}$$

The model we have described has a bounded state-space, and for any given state j, the probability of an increase in state is zero. It is therefore trivial to show that the *absorption probability of such a system must be unity* [KARL 75]. This rather intuitive result suggests that, as expected, the mean CRI duration T_C will always be finite, regardless of our choice of model parameters. Employing the "summation of steps" technique from [KARL 75], we obtain the analytical expression for the mean CRI duration, given an initial state $j = N$,

$$T_C = \sum_{j=1}^{N} \frac{1}{P_S(j)} \tag{4.8}$$

The model described is a pure death process, with no possibility of increase from any state j. This explains the very simple form of equation (4.8) - it is merely a sum of the average sojourn times in each of the states the system descends down through, from $j=N$ to $j=1$.

A slightly more useful method of obtaining T_C is numerical recursion: a numerical recursive solution of the set of equations (4.5) - (4.7) will yield the probabilities $P(0, t)$, (t=1, 2, 3, …) from which T_C is obtained,

$$T_C = E[\phi] = \sum_{t=1}^{\infty} t \cdot P(0,t) . \tag{4.9}$$

Note that in all our numerical solutions, we use the termination condition: $t \cdot P(0, t) < 10^{-9}$. The method of numerical recursive solution of the model's state transition equations, is extremely useful because it yields the series of exact system occupancy probability distributions, from timeslots 1 through to ∞ (in theory). This then allows other important statistics to be calculated, such as the average background traffic offered during the mean CRI (important in the more complex models we consider, in Sections 4.2 through to 4.4), and the entire discrete probability density function for ϕ. However, if we were solely interested in calculating the mean CRI length, T_C, adopting an analytical solution approach would have been quicker and simpler.

Having determined T_C, we now know the maximum achievable throughput of our basic deadlock model: it is merely the ratio of N request messages to the T_C slots it takes to clear them from the contention state. Let us define this ratio as L_{crit}, signifying *critical load*. This load is considered critical because if the N requests arrive with a period less than T_C, the arrival rate exceeds the system's service rate, and the system becomes unsteady, with the number of outstanding requests increasing towards infinity. Note we do not say unstable because we have already proved analytically that T_C is always finite; in this sense unsteady signifies an uncontrolled increase in the number of backlogged requests, so that T_C, although theoretically finite, becomes so large that for practical purposes it tends to infinity. In situations where L_{crit} is very small, it may be desired to increase it by some other means. This is where *signalling capacity allocation schemes* play a role. Such schemes describe how to manage the CMS minislots, and three new schemes are proposed in Section 4.5.

4.2 Bernoulli (BER) Deadlock Model

We now look at another important set of models - those which in addition to errored CMS's, take into account the possibility of new request arrivals during the contention resolution phase of an initial batch of N outstanding message requests. It is important for a model to capture the complete set of circumstances which may be part of a deadlock situation; and, with the models we are going to present in this and the following two sections, we capture not only the impact of collided retry requests using the p-persistence CRA, but at the same time highlight the effect of new request arrivals on stability and average length of the CRI. The work in [BISD 96c] makes an important contribution in a similar area of study for HFC MAC protocols, employing tree-based CRAs, and the contribution of our work adds to this body of knowledge by considering the useful and inherently simple p-persistence CRA.

The description of the base properties of the BER model and the notation used, are the same as those in Section 4.1, with the main difference being that now there is a possibility of a single additional arrival during the current CMS minislot, with a probability P_{arr} (in addition to any given CMS having the possibility of suffering a random error, with probability P_{err}). The probability P_{arr} is constant and therefore independent of the system state, j. Thus the label *Bernoulli* background traffic model (BER) arises, alluding to the fact that each new CMS, regardless of current system state, is effectively a Bernoulli trial with only two possible outcomes - a new request arrival with probability P_{arr}, and no new arrival with

probability (1-P_{arr}). The fact that the BER model only caters for Bernoulli outcomes of one or zero new arrivals, causes the mean arrival rate to equal the probability of a new arrival; as seen in the state-independent expression below, we label this quantity λ,

$$E[arr, j \in 0,1,\ldots\infty]^{BER\_\mathrm{mod}\,el} = P_{arr} \cdot 1 = \lambda \tag{4.10}$$

It is assumed that initially at t=1, N stations out of the potentially *infinite* pool (i.e. our system has an unbounded state space) of stations have a single-cell message to transmit. One should recall that our aim is to investigate worst-case traffic scenarios leading to contention channel deadlock, and we therefore assume all stations only transmit the worst-case type of traffic as far as signalling is concerned - ***single-cell messages***. When a station's traffic only comprises single-cell messages, it means that a separate contention request needs to be generated per every cell transmitted, as opposed to, say, one in ten cells for ten-cell messages.

Furthermore, we assume a worst-case arrival pattern of these messages at each station, so that after one such message arrives and is served, the station's queue once again returns to empty and there is no opportunity for utilising the contention-free "piggybacking" feature of most HFC and wireless ATM access protocols. This kind of behaviour may be seen in a scenario where for example, a large number of stations (say N) each generate a single message simultaneously, in what could be termed a "synchronised signalling storm" event.

Practically, the worst-case message length and arrival patterns just described, mean that during one contention resolution interval, a given station may only undergo transitions between two states: the *Inactive State* - the station has no single-cell messages to send and is therefore considered inactive and potentially ready to generate a new request at any time; and the *Active State* - the station has a single message to send and is contending for the channel, thus contributing exactly one outstanding request to the system-wide count, denoted by j. An important observation that needs to be made here is that theoretically, for arbitrary size messages, there is also a third possible state in which the stations may find themselves: the "data transmission in progress" state. For example, a station transmitting a long (e.g. 30 cell) message is neither contending for the channel, so it is not in the Active State; nor can it be said to be ready to generate a new request at any time, so it is not in the Inactive State. However, our single-cell message assumption means that no station can spend more than the almost negligible transmission time of a single cell in this "data transmission in progress" transient state. Thus, for practical purposes the existence of this third state may be ignored, thereby simplifying analysis of the problem.
After the first time all N stations try to access with probability p, but there is now also the potential either for an error or an additional request arrival to happen in any given timeslot, with probability P_{err} and P_{arr} respectively. For the number of outstanding requests to be reduced by one ("success"), we need there to be no new arrival, no CMS error, and only one of the existing requests to retry gaining access to the channel. The probability of a successful transmission is therefore given by $(1-P_{err})\cdot(1-P_{arr})\cdot Np(1-p)^{N-1}$. In general, when we have j stations contending for transmission, the probability of success (i.e. a reduction of one in the number of requests "waiting" for resolution) will be given by

$$P_S(j) = (1 - P_{err}) \cdot (1 - P_{arr}) \cdot jp(1-p)^{j-1} \tag{4.11}$$

The failure outcome when j stations are contending is no longer defined as a single event, but can be divided into the "no change" and "increase" events. The former event, resulting in no change of state, occurs either when (i) an arrival occurs, no existing requests retry, and the CMS is error-free, or (ii) when no arrival occurs and we do not get only one existing request trying to seize a CMS (this latter outcome means either a collision has occurred, or none have tried to seize the CMS), regardless of the CMS's error status; or, finally, when a single existing request retries, in the absence of an arrival but in the presence of an errored CMS. This probability is therefore given by

$$\begin{aligned} P_{NoCh}(j) = {} & P_{arr} \cdot (1 - P_{err}) \cdot (1-p)^j \\ & + (1 - P_{arr}) \cdot (1 - jp(1-p)^{j-1}) \\ & + (1 - P_{arr}) \cdot P_{err} \cdot jp(1-p)^{j-1} \end{aligned} \tag{4.12}$$

The "increase" event will happen either when (i) an additional request arrival occurs at the same time that one or more existing requests retry to seize a CMS, regardless of whether an error occurs or not, or when (ii) an additional request arrives, but is errored, and no existing requests retry to seize a CMS. The probability is then given by

$$P_{Inc}(j) = P_{arr} \cdot (1 - (1-p)^j) + P_{arr} \cdot P_{err} \cdot (1-p)^j \tag{4.13}$$

Unlike in the case of the basic model, where $j=N$ was an upper bound on the system occupancy and hence a boundary condition, the state can now rise to ∞ due to the possibility of further arrivals (without any limit, since we have an infinite pool of potential request generators). Therefore, when j=2, 3, 4.... ∞ we have

$$\begin{aligned} P(j,t) = {} & P(j+1, t-1) \cdot P_S(j+1) \\ & + P(j, t-1) \cdot P_{NoCh}(j) \\ & + P(j-1, t-1) \cdot P_{Inc}(j-1) \end{aligned} \tag{4.14}$$

Similar boundary conditions as for the base model need to be put in place, since the j=1 and j=0 states are again special cases. When j=1, there is no possibility of having reached this state from the absorbing j=0 state in the previous timeslot. So the state transition equation for j=1 is,

$$\begin{aligned} P(j,t) = {} & P(j+1, t-1) \cdot P_S(j+1) \\ & + P(j, t-1) \cdot P_{NoCh}(j) \end{aligned} \tag{4.15}$$

On the other hand, if a system is in the absorbing state j=0 at time t, then it could only have arrived by descending from the j=1 state in the time period [*t-1*, *t*], as the transition equation for j=0 shows,

$$P(j,t) = p \cdot P(1, t-1) \cdot (1 - P_{arr}). \tag{4.16}$$

4.2.1 Calculating the Probability of Absorption into State 0

In order to calculate the probability of absorption into state 0, we adopt a similar approach and notation used in [KARL 75], but keep in mind that here we are dealing with a discrete-time Markov chain (rather than a continuous-time Birth Death process). Nonetheless, simple manipulation of some steps used in deriving Theorem 7.1 in [KARL 75] shows that the end result is exactly the same, whether we are dealing with a continuous- or discrete-time Birth Death process.

We start by defining the *birth* and *death* rates for the Markov chain by $\Pi_{up}(i) = P_{Inc}(i) / \left(P_{Inc}(i) + P_S(i)\right)$ and $\Pi_{dn}(i) = P_S(i) / \left(P_{Inc}(i) + P_S(i)\right)$ respectively, in line with our original notation for probabilities of state increase and decrease. These rates may be interpreted as conditional probabilities of the state transitions $i \to i+1$ (up) and $i \to i-1$ (down), given that a transition out of state *i* does occur. It is also to be observed that $\Pi_{dn}(i) + \Pi_{up}(i) = 1$, since by definition of the BER model, these two transitions are the only ones possible out of state *i*.

As in [KARL 75] we let u_i denote the probability of being absorbed into state 0, from the initial state *i*. Theorem 7.1 from [KARL 75] may hence be written as,

$$u_i = \begin{cases} \dfrac{\displaystyle\sum_{j=i}^{\infty}\left(\prod_{k=1}^{j}\left(\Pi_{dn}(k)/\Pi_{up}(k)\right)\right)}{1+\displaystyle\sum_{j=1}^{\infty}\left(\prod_{k=1}^{j}\left(\Pi_{dn}(k)/\Pi_{up}(k)\right)\right)}, & \text{if} \quad \displaystyle\sum_{j=1}^{\infty}\left(\prod_{k=1}^{j}\left(\Pi_{dn}(k)/\Pi_{up}(k)\right)\right) < \infty \\ 1 & \text{if} \quad \displaystyle\sum_{j=1}^{\infty}\left(\prod_{k=1}^{j}\left(\Pi_{dn}(k)/\Pi_{up}(k)\right)\right) = \infty \end{cases} \tag{4.17}$$

4.2.2 Calculating the "Practical" Average Length of the CRI, T_c

The mean time to absorption may be obtained either (i) by using a familiar closed-form analytical expression applicable to discrete-time Birth-Death Markov chains of this type [KARL 75]; or, (ii) by using numerical recursive methods to solve the state transition equations (4.14) through (4.16) and then extracting T_C from the known probabilities $P(0, t)$, (t= 1, 2, 3, …). These probabilities together form the discrete-time probability density function of the random variable ϕ which represents the absorption time (i.e. the length of the CRI).

Let us first turn our attention to the analytical method. When the parameter set *(L, N, p, Perr, λ)* is such that equation (4.17) yields a probability of exactly 1, the mean time to absorption is finite and we can simply apply the following formula, again from Theorem 7.1 in [KARL 75] to obtain T_C,

$$T_C = \sum_{j=1}^{\infty} \rho_j + \sum_{r=1}^{i-1} \left(\left(\prod_{k=1}^{r} \left(\Pi_{dn}(k) / \Pi_{up}(k) \right) \right) \cdot \sum_{j=r+1}^{\infty} \rho_j \right), \qquad if \quad \sum_{j=1}^{\infty} \rho_j < \infty \tag{4.18}$$

where $\rho_j = (\Pi_{up}(1)\Pi_{up}(2)\cdots\Pi_{up}(j-1)) / (\Pi_{dn}(1)\Pi_{dn}(2)\cdots\Pi_{dn}(j))$ and $\Pi_{up}(k)$, $\Pi_{dn}(k)$ are as defined in Section 4.2.1. On the other hand, when the system parameters are such that $\sum_{j=1}^{\infty} \rho_j = \infty$ because the probability of absorption into state 0 is less than 1, [KARL 75] explains that the theoretical mean time to absorption must be infinite. The BER Deadlock model which we are considering has no upper bound on state occupancy, and so there is always a finite probability that absorption will not occur, as per equation (4.17). That is, the BER model theoretically satisfies the condition $\sum_{j=1}^{\infty} \rho_j = \infty$, which causes T_C to always be infinite, and the BER model to be theoretically unstable for any background arrival rate λ.

From the foregoing discussion, it becomes clear that we need a measure for the *practical* mean CRI length: that is, the value of T_C given that the probability of non-absorption is sufficiently small, that we can assume the BER model to be practically stable. The test for practical stability in this case is simple: if the absorption probability of equation (4.17) is greater than 1-ε, where ε is our accuracy threshold (10^{-9}), the BER modelled system is treated as practically stable and a practically finite value of T_C may be computed. Otherwise, we state that the system is unstable and that a finite value of T_C is not feasible.

The same observations are made if we attempt to use the numerical method of finding T_C by applying the standard mean formula on the ϕ p.d.f. That is, the value of T_C given numerically by summing over the [1,∞] range of t values,

$$E[\phi] = T_C(L, N, p, P_{err}, \lambda) = \sum_{t=1}^{\infty} t \cdot P(0,t) \tag{4.19}$$

will either be practically finite, when the absorption probability is "practically one" (>1-ε); or, it will be infinite and impossible to calculate using this approach, when the probability of absorption into state zero is less than 1-ε.

4.2.3 Obtaining the Critical Load, L_{crit}

Recall that a vital measure of any CRA is the maximum achievable signalling throughput, which we earlier termed the critical load, or L_{crit}. Unlike the simple expression for the basic deadlock model, here we can see that: if it takes on average T_C timeslots to clear an initial batch of N requests (in the presence of background traffic), then the critical load can be thought of as a direct measure of how much total traffic (i.e. not just the initial batch of N requests) can be carried during the T_C timeslots it takes to clear the original batch.

The expression below captures this statement for the BER model,

$$L_{crit} = \frac{N + E[arr\_during\_T_C]^{BER\_model}}{T_C} \tag{4.20}$$

As we are dealing with a model where the mean arrival rate is constant and does not vary with changing system state occupancy distribution (i.e. it is equal to λ for all j), the mean number of background arrivals during T_C is simply given by

$$E[arr\_during\_T_C]^{BER\_model} = T_C \cdot \lambda \tag{4.21}$$

4.3 "Machine Service" (MSV) Deadlock Model

The MSV model is wholly based on the description we have given for the BER model in Section 4.2, with the primary differences being: (i) a limited set of stations, with population L; and, (ii) a state-dependent rather than constant probability of an additional single arrival during a timeslot, denoted by $P_{arr}(j)$. As per the BER model, we assume single-cell messages and no use of piggybacked slots. Also, the existence of only two states in which the stations could be in is assumed: *Active* and *Inactive*, as defined in Section 4.2.

The term MSV model, refers to the similarity of this state-dependent arrival probability, to that of the well-known "Machine-SerVice" model [TAHA 76], where there is a finite number of sources ("machines") and servers ("machine repairmen"). Importantly, as in the original MSV model the key assumption is that we have a finite station population, L. It is intuitive that any expression for $P_{arr}(j)$ must reflect that the probability of an arrival decreases as the number of inactive stations decreases, since there are less potential arrival generators. There are L-j inactive stations (i.e. those in the Inactive State) when we have j outstanding message requests (j stations in the Active State). The following expression for $P_{arr}(j)$ is used when the system is in state j,

$$P_{arr}(j) = \begin{cases} \lambda \cdot \dfrac{L-j}{L}, & j = 0,1,2,\ldots(L-1) \\ 0 \qquad\quad , & j = L \end{cases} \tag{4.22}$$

where λ is the probability <u>of a message request arrival for the system</u> (i.e. not merely the probability that one station changes from Inactive to Active state), during a single timeslot. The notation used (λ) is meant to reinforce that for the purposes of comparison with the infinite-station BER model, we match the parameters so that the mean arrival rate of the MSV model when j=0, is the same as the state-independent mean arrival rate of the BER model, or $E[arr, j=0]^{MSV\_model} = \lambda \cdot \left((L-0)/L\right) = \lambda = E[arr, j \in 0,1,\ldots\infty]^{BER\_model}$. This relationship signifies that with all stations Inactive and able to generate a request, the mean request arrival rate (average number of requests per unit timeslot) into the system is equal for the MSV model, to that for the BER model. As more stations migrate to the Active State, the subsequent reduction in the number of potential request generators causes a linearly proportional drop in the mean request arrival rate.

This type of parameter matching suggests the inequality

$$E[arr,\tau]^{MSV\_model} = \sum_{j=1}^{L} P(j,\tau)\cdot E[arr,j]^{MSV\_model} \quad < \quad \lambda \tag{4.23}$$

with the per-timeslot mean arrival rate averaged, at some arbitrary timeslot $t=\tau$, over an arbitrarily occupied transient state space [1, L] (the notion of a transient set of states is discussed in Section 4.3.3). Regardless of the particular state occupancy distribution at $t=\tau$, the relationship $E[arr,\tau]^{MSV\_model} < \lambda$ still holds. This is observed since by virtue of equation (4.22), the MSV model's mean arrival rate is always less than or equal to the BER model's constant, state-independent mean arrival rate λ. Section 4.3.3 discusses the impact of $E[arr,\tau]^{MSV\_model}$ on L_{crit}, the system's critical signalling load.

Note the boundary condition of equation (4.22), which highlights that $P_{arr}(j)$ is zero when $j=L$ and there are no more inactive stations to generate new arrivals. It is important to realise that although $j=0$ represents the absorbing state, $P_{arr}(0)$ is not zero. The reason for this lies in our formulation of the problem, whereby we desire to calculate the mean length of a single contention resolution interval, T_C, which by definition ends when the number of outstanding requests returns, from some initial state $j=N$, to zero for the first time. This means that for the purposes of calculating T_C, it is impossible for the system to make a transition out of the absorbing state $j=0$, *but this does not mean that the probability of arrival in that state is also zero* - in fact, it is at its maximum value λ, since all L stations are potential request generators when $j=0$.

Assume that initially at $t=1$, N out of the L stations have a single-cell message to transmit. After the fist timeslot, all N of these stations try to access with probability p, but there is also the potential either for an error or an additional request arrival to happen in any given timeslot, with probability P_{err} and $P_{arr}(j)$ respectively. For the number of outstanding requests to be reduced by one ("success"), we need there to be no new arrival, no CMS error, and only one of the existing requests to retry gaining access to the channel. The probability of successful transmission is therefore given by $(1-P_{err})\cdot(1-P_{arr}(N))\cdot Np(1-p)^{N-1}$. In general, when we have j stations contending for transmission, the probability of success (i.e. a reduction of one in the number of requests "waiting" for resolution) will be given by

$$P_S(j) = (1-P_{err})\cdot(1-P_{arr}(j))\cdot jp(1-p)^{j-1} \tag{4.24}$$

As for the BER model, the failure outcome when j stations are contending is no longer defined as a single event, but can be divided into the "no change" and "increase" events. The former event occurs either when (i) an arrival occurs, no existing requests retry, and the CMS is error-free, or (ii) when no arrival occurs and we do not get only one existing request trying to seize a CMS (this latter outcome means either a collision has occurred, or none have tried to seize the CMS), regardless of the CMS's error status; or, finally, (iii) when a single existing request retries, in the absence of an arrival but in the presence of an errored CMS.

This composite probability is therefore given by

$$\begin{aligned} P_{NoCh}(j) &= P_{arr}(j)\cdot(1-P_{err})\cdot(1-p)^{j} \\ &+ (1-P_{arr}(j))\cdot(1-jp(1-p)^{j-1}) \\ &+ (1-P_{arr}(j))\cdot P_{err}\cdot jp(1-p)^{j-1} \end{aligned} \tag{4.25}$$

The "increase" event will happen either when (i) an additional request arrival occurs at the same time that one or more existing requests retry to seize a CMS, regardless of whether an error occurs or not, or when (ii) an additional request arrives, but is errored, and no existing requests retry to seize a CMS. The probability is then given by

$$P_{Inc}(j) = P_{arr}(j)\cdot\left((1-(1-p)^{j}) + P_{err}\cdot(1-p)^{j}\right) \tag{4.26}$$

Note that for any j, knowing any two of the three probabilities $P_{Inc}(j)$, $P_S(j)$, and $P_{NoCh}(j)$, will yield the third since necessarily $P_{Inc}(j)+P_S(j)+P_{NoCh}(j)=1$.

Since we are dealing with a limited-station model, the state space has an upper bound at $j=L$. This gives rise to a number of special boundary conditions, when setting up the state transition equations:

Namely, for $j=0$,

$$P(j,t) = P_S(j+1)\cdot P(j+1,t-1)\,. \tag{4.27}$$

since there is no way of reaching the absorbing state $j=0$, other than from the state $j=1$.

The boundary condition for $j=1$, is such that

$$\begin{aligned} P(j,t) &= P_S(j+1)\cdot P(j+1,t-1) \\ &+ P_{NoCh}(j)\cdot P(j,t-1) \end{aligned} \tag{4.28}$$

since the state $j=1$ may not be reached from the absorbing state $j=0$.

The last special case is that of $j=L$, where the transition equation becomes,

$$\begin{aligned} P(j,t) &= P(j,t-1)\cdot P_{NoCh}(j) \\ &+ P(j-1,t-1)\cdot P_{Inc}(j-1) \end{aligned} \tag{4.29}$$

because it is impossible to be in any state $j>L$, and hence it is impossible to "descend" to the state $j=L$ from any higher state.

If the system is not in any of the three special-case states we have just covered, the most general state transition equation applies,

$$\begin{aligned} P(j,t) = & P_S(j+1)\cdot P(j+1,t-1) \\ & + P_{NoCh}(j)\cdot P(j,t-1) \\ & + P_{Inc}(j-1)\cdot P(j-1,t-1) \end{aligned} \tag{4.30}$$

4.3.1 Calculating the Probability of Absorption into State 0

Let u_i denote the probability of being absorbed into state 0, from the initial state *i*. We adopt a similar approach and notation used in [KARL 75], but keep in mind that here we are dealing with a discrete-time Markov chain (rather than a continuous-time Birth Death process). Thus, a recursion formula for u_i may be obtained by considering the only possible states after one timeslot has passed. Namely, as we go from *t = t'* to *t = t'+1*, the following events are possible:

$$i \to i+1 \qquad \textit{with probability} \quad P_{Inc}(i)\,, \tag{4.31}$$

$$i \to i \qquad \textit{with probability} \quad P_{NoCh}(i), \tag{4.32}$$

$$\text{and} \qquad i \to i-1 \qquad \textit{with probability} \quad P_S(i)\,. \tag{4.33}$$

Using these event probabilities, and considering the passage of one time unit, we obtain:

$$u_i = 1 \quad \textit{by definition} \qquad \textit{for} \quad i = 0, \tag{4.34}$$

$$\text{and} \qquad u_i = P_{Inc}(i)\cdot u_{i+1} + P_S(i)\cdot u_{i-1} + (1 - P_{Inc}(i) - P_S(i))\cdot u_i \quad \textit{for} \quad 1 \le i < L\,. \tag{4.35}$$

Rearranging (4.35), we obtain a general equation of the same form used in [KARL 75]:

$$u_i = \Pi_{up}(i)\cdot u_{i+1} + \Pi_{dn}(i)\cdot u_{i-1} \qquad \textit{for} \qquad 1 \le i < L\,. \tag{4.36}$$

where $\Pi_{up}(i)$ and $\Pi_{dn}(i)$ are the birth and death rates for the Markov chain under consideration, and have been already defined in Section 4.2.1. It is also to be observed that $\Pi_{dn}(i) + \Pi_{up}(i) = 1$, since by definition of the MSV model, these two transitions are the only ones possible out of state *i*. In equation (4.36) the focus is no longer on the probability of absorption after one timeslot, but after one explicit transition out of state *i*.

Note that we have a special case when *i=L*, since the following relations hold:

$$\Pi_{up}(L) = \frac{P_{Inc}(L)}{P_{Inc}(L) + P_S(L)} = 0 \qquad (\text{since} \quad P_{Inc}(L) = 0\,) \tag{4.37}$$

$$\text{and} \qquad \Pi_{dn}(L) = \frac{P_S(L)}{P_{Inc}(L) + P_S(L)} = 1 \qquad (\text{since} \quad P_{Inc}(L) = 0\,) \tag{4.38}$$

Substituting (4.37) and (4.38) into the general recursive equation (4.36), when $i=L$ we obtain,

$$u_L = u_{L-1}. \tag{4.39}$$

Continuing, we then back-substitute (4.39) into equation (4.36) again for $i=L-1$, and get

$$\begin{aligned} &u_{L-1} = \Pi_{up}(L-1)\cdot u_L + \Pi_{dn}(L-1)\cdot u_{L-2} \\ &u_{L-1}\cdot(1-\Pi_{up}(L-1)) = \Pi_{dn}(L-1)\cdot u_{L-2} \\ &u_{L-1}\cdot\Pi_{dn}(L-1) = \Pi_{dn}(L-1)\cdot u_{L-2} \\ &\therefore u_{L-1} = u_{L-2} \end{aligned} \tag{4.40}$$

It is then easy to see that continuing this pattern of back-substitution, we obtain the equation

$$u_L = u_{L-1} = u_{L-2} = \ldots\ldots = u_2 = u_1 = u_0 = 1 \tag{4.41}$$

which explicitly tells us that for this finite state-space, limited-source MSV arrival model, *the probability of absorption into state 0 is always 1, regardless of the initial state i.* This is a somewhat intuitive result, because the finite state-space is divided into a set of *L-1* transient states (which may only be "visited" a finite number of times) and one recurrent state (which is the absorbing state 0, which, when reached, is "visited" an infinite number of times).

4.3.2 Calculating the Average Length of the CRI T_c (or, Mean Time to Absorption)

4.3.2.1 Numerical Recursion

A numerical recursive solution of the state transition equations for the MSV model, (4.27) through (4.30), will give the probabilities $P(0, t)$, (t=1, 2, 3, …). The set of $P(0, t)$ for all t, is the discrete-time probability density function of the absorption time, ϕ. As for the BER model, the average of ϕ then yields the mean time for the system to be absorbed into state 0, denoted by $T_C(L, N, p, Perr, \lambda)$,

$$T_C(L, N, p, P_{err}, \lambda) = \sum_{t=1}^{\infty}\left(t\cdot P(0,t)\right) \tag{4.42}$$

The expression in (4.42) shows T_c to be a function of the type of request arrival model used (L, λ), the p-persistence parameter (p), the CMS error probability (P_{err}), as well as the batch size (N).

In our numerical solutions presented in the next section we have used the termination condition: $t\cdot P(0, t) < \varepsilon$ with $\varepsilon = 10^{-9}$. It is important to realise that equation (4.42) will only give a valid

approximation to the true value of T_c, if the conditions are such that the *"practical" probability of absorption (PPA)* into state 0 is very close to 1:

$$PPA = \sum_{t=1}^{T_{pr}} P(0,t) \quad \geq \quad 1-\varepsilon \tag{4.43}$$

From its definition in (4.43), we see that *PPA* is the probability that absorption into state 0 has already taken place by some very large value of $t = T_{pr}$, which would be equivalent to infinity for the practical purposes of HFC or Wireless ATM contention resolution algorithms. An example is the value of T_{pr} = 12,500,000 timeslots, which represents roughly 10 minutes of real-time in an implementation of the F-CPR protocol for HFC access networks, with 60-byte upstream slots and an upstream channel speed of 10 Mbit/s. Any situation where all outstanding messages cannot be cleared by T_{pr} (i.e. absorption into state 0 does not take place by such time) would undoubtedly be considered a catastrophic system deadlock for any contention resolution algorithm.

4.3.2.2 Analytical Solution

Given that equation (4.41) holds, and absorption is certain from any initial state *i*, let ω_i denote the mean time taken to be absorbed into state 0, from the initial state *i*. As in 4.3.1, a recursion formula for ω_i may be obtained by firstly considering the only possible states after the first timeslot has passed. Using the definitions of (4.31) - (4.33) for the event probabilities, the following equations hold:

$$\omega_i = 0 \qquad \textit{by definition} \qquad \textit{for} \qquad i = 0, \tag{4.44}$$

and $\omega_i = 1 + P_{Inc}(i)\cdot\omega_{i+1} + P_S(i)\cdot\omega_{i-1} + (1 - P_{Inc}(i) - P_S(i))\cdot\omega_i \quad \textit{for} \quad 1 \leq i < L\,.$ (4.45)

Rearranging (4.45), we once again get a general equation of the same form used in [KARL 75]:

$$\omega_i = \left(\frac{1}{P_{Inc}(i) + P_S(i)}\right) + \Pi_{up}(i)\cdot\omega_{i+1} + \Pi_{dn}(i)\cdot\omega_{i-1} \quad \text{for} \qquad 1 \leq i < L\,. \tag{4.46}$$

The case $i=L$ is a special case again, for the same reason as in 4.3.1. Thus, substituting (4.37) and (4.38) into the general recursive equation (4.46), when $i = L$ we obtain,

$$\xi_{L,L-1} = \omega_L - \omega_{L-1} = \frac{1}{P_S(L)}\,. \tag{4.47}$$

Further back-substitution of (4.47) into equation (4.46) for $i=L$-1, yields

$$\begin{aligned}
\omega_{L-1} &= \left(\frac{1}{P_{Inc}(L-1)+P_S(L-1)}\right)+\Pi_{up}(L-1)\cdot\omega_L+\Pi_{dn}(L-1)\cdot\omega_{L-2} \\
\omega_{L-1}\cdot(\Pi_{up}(L-1)+\Pi_{dn}(L-1)) &= \left(\frac{1}{P_{Inc}(L-1)+P_S(L-1)}\right)+\Pi_{up}(L-1)\cdot\omega_L+\Pi_{dn}(L-1)\cdot\omega_{L-2} \\
\Pi_{dn}(L-1)\cdot(\omega_{L-1}-\omega_{L-2}) &= \left(\frac{\Pi_{dn}(L-1)}{P_S(L-1)}\right)+\Pi_{up}(L-1)\cdot(\omega_L-\omega_{L-1}) \\
(\omega_{L-1}-\omega_{L-2}) &= \left(\frac{1}{P_S(L-1)}\right)+\frac{P_{Inc}(L-1)}{P_S(L-1)}\cdot(\omega_L-\omega_{L-1}) \\
\therefore \xi_{L-1,L-2} &= \left(\frac{1}{P_S(L-1)}\right)+\frac{P_{Inc}(L-1)}{P_S(L-1)}\cdot\xi_{L,L-1}
\end{aligned} \tag{4.48}$$

Continuing this pattern of back-substitution, the general expression for the mean time to first make the transition $i \to i-1$ becomes

$$\xi_{i,i-1} = \left(\frac{1}{P_S(i)}\right)+\frac{P_{Inc}(i)}{P_S(i)}\cdot\xi_{i+1,i} \quad \textit{for} \quad 1 \le i < L\,. \tag{4.49}$$

so that the mean time to absorption into state 0, from initial state $i=N$, is just a sum of the component steps

$$\omega_N = \sum_{i=1}^{N}\xi_{i,i-1}\,. \tag{4.50}$$

Equation (4.49) lends a lot of insight into the behaviour of ω_i (which can equivalently be interpreted as T_c for initial batch size of $i=N$, using our notation defined previously). We first observe that the probabilities of gaining or removing an outstanding message request in one timeslot, $P_{Inc}(i)$ and $P_S(i)$ respectively, have a significant impact on the behaviour of individual transitions' step durations. Firstly, we note that necessarily $\xi_{i,i-1} > \xi_{i+1,i}$ through the recursive nature of the equation defining the step durations. Now, if $P_{Inc}(i) < P_S(i)$ for all i, the step duration $\xi_{i,i-1}$ converges to a particular value as $i \to 1$. If on the other hand $P_{Inc}(i) > P_S(i)$ for all i, the step duration clearly diverges as $i \to 1$. The MSV model is a system where the relationship between $P_{Inc}(i)$ and $P_S(i)$ changes with state i, so that for small to moderate i we have $P_{Inc}(i) < P_S(i)$. For almost all useful parameter combinations, there is then a turning point at some i^* so that for $i \ge i^*$ we observe $P_{Inc}(i) > P_S(i)$. This leads to hybrid behaviour in terms of the step duration, with divergence over the range of states $[i^*, L]$, and convergence for states smaller than i^*. As will be highlighted in the results section, the fact that the step duration diverges when $i \ge i^*$, means that *PPA* tends to decrease to below 1-ε, for parameter combinations leading to $P_{Inc}(i) >> P_S(i)$. There is no closed form expression which gives the critical parameter values $\{L, N, p, Perr, \lambda\}$ for which the *PPA* first begins to fall below 1-ε, which is why the numerical treatment of this practically significant problem is an interesting one.

4.3.3 Obtaining the Critical Load, L_{crit}

As before, the critical load measures how much total traffic (the initial batch of N requests plus the additional background requests) can be carried during the T_c timeslots it takes to clear the original batch,

$$L_{crit} = \frac{N + E[arr\_during\_T_C]^{MSV\_model}}{T_C} \tag{4.51}$$

with the mean number of background arrivals during T_C given by

$$\begin{aligned} E[arr\_during\_T_C]^{MSV\_model} &= \sum_{t=1}^{\infty} P(0,t) \cdot \left(\sum_{\tau=1}^{t} E[arr,\tau]^{MSV\_model} \right) \\ &= \sum_{t=1}^{\infty} P(0,t) \cdot \left(\sum_{\tau=1}^{t} \left(\sum_{j=1}^{L} P(j,\tau) \cdot E[arr,j]^{MSV\_model} \right) \right) \\ &= \sum_{t=1}^{\infty} P(0,t) \cdot \left(\sum_{\tau=1}^{t} \left(\sum_{j=1}^{L} P(j,\tau) \cdot \frac{L-j}{L} \cdot \lambda \right) \right) \end{aligned} \tag{4.52}$$

Equation (4.52) is not nearly as simple as (4.21), where the mean arrival rate was a constant, state-invariant value of λ requests per timeslot. Instead, (4.52) is essentially a compounded set of three weighted means. In the innermost, we evaluate the mean number of background arrivals for a given timeslot, τ. Note that this depends on both the likelihood of occupying a certain state, as well as the average arrival rate when in that state. The second summation totals up all contributions for τ, from the initial timeslot 1, right up to the time t. This sum represents the total (average) background request arrivals in the time interval [1, t]. The final step is to perform a weighted average of these totals, for all values of t from 1 to ∞, similarly to the way in which we calculate the average CRI length, T_C, in equation (4.42). This then results in the desired quantity, **the total (average) background request arrivals in the time interval [1, T_C].**

Another interesting feature of equation (4.52), is that the j=0 point in the state space is not valid for this calculation because of the absorbing nature of that state (i.e. when it is reached, we consider the CRI finished). This means that although the probability of request arrivals is non-zero while in that state, the probability that these arrivals lead to an exit from that state remains zero. That is why we term the [1, L] region of the state space transient - any state is reachable from the transient region, including the absorbing state j=0. It is important to realise that, if the system's actual signalling load exceeds L_{crit}, as given by (4.51), a state of **signalling deadlock** is encountered, where the inability of the signalling messages to get through causes back-pressure and eventual buffer overflow and data loss at the end stations. That is, if we think of all the unresolved requests as belonging to a virtual queue, which is bounded by the total station population size L, then this queue would have a service rate L_{crit} requests per timeslot.

Any situation where this service rate is exceeded for long periods of time would result in the queue occupancy increasing to its maximum size, *L*, and hovering in, or very close, to *L* requests thereafter. As per the discussion in Section 4.3.2.2, although it is theoretically possible to exit from state *L*, the total average time for even for the smallest transitions (e.g. $L \to L-1 \to L-2$) is for all practical purposes and useful parameter values considered (in Chapter 6), an infinitely large number of timeslots (in real-time terms, for the parameters considered, equivalent to a timescale ranging from hours to years!).

4.4 Binomial (BIN) Deadlock Model

The notation used, as well as all of the model assumptions and characteristics are the same as in section 4.3, with the only difference between the MSV and Binomial (BIN) models being the different nature of the state-dependent arrival probability $P_{arr}(x, j)$. The Binomial model is so named because, during a unit timeslot, the probability distribution of the number of newly arriving message requests is Binomial, and so the $P_{arr}(x, j)$ function now needs another dimension, to represent how many new requests arrive. Note that the MSV model, where at most one new arrival was allowed per unit timeslot, becomes a special case of this more general BIN model. Using the BIN model, the only limit on the number of new arrivals is the physical limit of how many potential arrival generators currently exist. This physical limit is *L-j*, determined by total station population *L*, and the current system state *j*, being the number of active stations with one outstanding request each. The expression for $P_{arr}(x, j)$ is thus

$$P_{arr}(x,j) = \begin{cases} \binom{L-j}{x} \cdot \left(\frac{\lambda}{L}\right)^x \cdot \left(1-\left(\frac{\lambda}{L}\right)\right)^{L-j-x}, & j = 0,1,2,..(L-1); \quad x = 0,1,...,L-j \\ 0 \qquad\qquad\qquad\qquad , & j = L \end{cases} \tag{4.53}$$

where (λ/L) represents the probability of a single station changing from the Inactive to the Active State during a single timeslot, *assumed independent of any other event*. Following our approach with the MSV model, in order to enable comparison with the infinite-station BER model, we choose the parameters so that $E[arr, j=0]^{BIN\_model} = L \cdot \left(\frac{\lambda}{L}\right) = \lambda = E[arr, j \in 0,1,...\infty]^{BER\_model}$. This relationship signifies that with all stations Inactive and able to generate a request, the mean request arrival rate per timeslot is equal for the MSV model, to that for the BER model. As more stations migrate to the Active State, the subsequent reduction in the number of potential request generators causes a linearly proportional drop in the mean request arrival rate. As in the case of the MSV model previously, this type of parameter matching suggests that

$$E[arr, \tau]^{BIN\_model} = \sum_{j=1}^{L} P(j, \tau) \cdot E[arr, j]^{BIN\_model} \quad < \quad \lambda \tag{4.54}$$

with $E[arr, j]^{BIN\_model}$ averaged, at some arbitrary timeslot $t=\tau$, over an arbitrarily occupied transient state space [1, *L*]. In other words, regardless of the particular state occupancy distribution at $t=\tau$, the relationship $E[arr, \tau]^{BIN\_model} < \lambda$ will hold. This inequality stems from equation (4.53) - due to which

the BIN model's mean arrival rate is always less than or equal to the BER model's constant and state-independent mean arrival rate λ. Section 4.4.3 quantifies the impact of $E[arr,\tau]^{BIN\_\mathrm{mod}el}$, on L_{crit}, the system's critical signalling load.

The boundary condition of equation (4.53), which highlights that $P_{arr}(x, j)$ is zero for all x, when $j=L$ and there are no more inactive stations to generate new arrivals. Importantly, although $j=0$ represents the absorbing state, $P_{arr}(x, 0)$ is non-zero for all x. As was the case with the MSV model, the reason for this lies in our formulation of the problem. Even though it is impossible for the system to make a transition out of the absorbing state $j=0$ for the purposes of calculating the average CRI length, T_c, *the Binomial probability distribution of x request arrivals while in that state is not zero.* In fact, since all L stations are potential request generators when $j=0$, the Binomial arrival distribution has the greatest mean value at this point, so that the $j=0$ mean arrival rate is $E[arr, j=0]^{BIN\_\mathrm{mod}el} = \lambda$ (see equation (4.53)).

As in the case of the MSV model, it is assumed that initially, at $t=1$, N out of the L stations have a single-cell message to transmit. After the fist timeslot, all N of these stations try to access with probability p, but there is also the potential either for an error and/or for x additional request arrivals to happen in any given timeslot, with probability P_{err} and $P_{arr}(x, j)$ respectively. For the number of outstanding requests to be reduced by one ("success"), we need there to be no new arrivals, no CMS error, and only one of the existing requests to retry gaining access to the channel. The probability of successful transmission is therefore given by $(1-P_{err})\cdot P_{arr}(0,N)\cdot Np(1-p)^{N-1}$. In general, when we have j stations contending for transmission, the probability of success (i.e. a reduction of one in the number of requests "waiting" for resolution) will be given by

$$P_S(j) = (1-P_{err})\cdot P_{arr}(0,j)\cdot jp(1-p)^{j-1} \tag{4.55}$$

The failure outcome when j stations are contending is no longer defined as a single event, but can be divided into the "no change" and "increase" events. The former event occurs either when (i) exactly one arrival occurs, no existing requests retry, and the CMS is error-free, or (ii) when zero arrivals occur and we do not get only one existing request trying to seize a CMS (this latter outcome means either a collision has occurred, or none have tried to seize the CMS), regardless of the CMS's error status; or, finally, (iii) when a single existing request retries, in the absence of an arrival but in the presence of an errored CMS. This probability is therefore given by the sum,

$$\begin{aligned} P_{NoCh}(j) = {} & P_{arr}(1,j)\cdot(1-P_{err})\cdot(1-p)^{j} \\ & + P_{arr}(0,j)\cdot(1-jp(1-p)^{j-1}) \\ & + P_{arr}(0,j)\cdot P_{err}\cdot jp(1-p)^{j-1} \end{aligned} \tag{4.56}$$

While the previous two outcomes of "no change" and "success" are identical for this BIN model, to those for the MSV model, the "increase" event now gains an added dimension, since it is possible to record an increase of more than just one request during a unit timeslot.

It is simple to see that as soon as more than one arrival is recorded in a timeslot ($x > 1$), regardless of all other event outcomes (retry or CMS error), the system state j will certainly increase by that number of arrivals. A special case arises when $x = 1$. Namely, it can be stated that an "increase of exactly $x=1$ requests" will happen either when (i) $x=1$ additional request arrival occurs at the same time that one or more existing requests retry to seize a CMS, regardless of whether an error occurs or not; or, when (ii) $x=1$ additional requests arrive, and no existing requests retry to seize a CMS, but the CMS is errored. The expression, conditional on the x value, is therefore given by

$$P_{Inc}(x, j) = \begin{cases} P_{arr}(x, j) \cdot \left((1-(1-p)^j) + P_{err} \cdot (1-p)^j\right), & x = 1 \\ P_{arr}(x, j), & x > 1 \end{cases} \tag{4.57}$$

The sum of the probabilities of all outcomes must by definition be equal to 1, so that

$$\sum_{x=1}^{L-j} P_{Inc}(x, j) = 1 - P_S(j) - P_{NoCh}(j) \tag{4.58}$$

The state space has an upper bound at $j=L$, due to the finite number of stations. This gives rise to very similar boundary conditions for the state transition equations, as we saw in section 4.3 for the MSV model.

When $j=0$,

$$P(j,t) = P_S(j+1) \cdot P(j+1, t-1)\,. \tag{4.59}$$

since there is no way of reaching the absorbing state $j=0$, other than from the state $j=1$.

The boundary condition for $j=1$, is such that

$$\begin{aligned} P(j,t) = {} & P_S(j+1) \cdot P(j+1, t-1) \\ & + P_{NoCh}(j) \cdot P(j, t-1) \end{aligned} \tag{4.60}$$

since the state $j=1$ may not be reached from the absorbing state $j=0$.

The last special case is that of $j=L$, where the transition equation becomes,

$$P(j,t) = P(j, t-1) \cdot P_{NoCh}(j) + \sum_{x=1}^{L-1} \left(P(j-x, t-1) \cdot P_{Inc}(x, j-x)\right) \tag{4.61}$$

because it is impossible to be in any state $j>L$, and hence it is impossible to "descend" to the state $j=L$ from any higher state. In a significant difference to the corresponding boundary condition of the MSV

model, it is now possible to "ascend" to the state $j=L$ in one unit of time, from any of the states between 1 and L-1, inclusive.

If the BIN-modelled system is not in any of the three special-case states we have just covered, the most general state transition equation applies,

$$\begin{aligned} P(j,t) &= P_S(j+1)\cdot P(j+1,t-1) \\ &+ P_{NoCh}(j)\cdot P(j,t-1) \\ &+ \sum_{x=1}^{j-1}\left(P_{Inc}(x,j-x)\cdot P(j-x,t-1)\right) \end{aligned} \tag{4.62}$$

4.4.1 Calculating the Probability of Absorption into State 0

As earlier, u_i denotes the probability of being absorbed into state 0, from the initial state i. A recursion formula for u_i may be obtained by considering the possible states after the first timeslot has passed. As we go from $t = t'$ to $t = t'+1$, the following events are possible:

$$i \to i+(L-i) \qquad \textit{with probability} \quad P_{Inc}(L\text{-}i,\, i)\,, \tag{4.63}$$

$$i \to i+(L-i-1) \qquad \textit{with probability} \quad P_{Inc}(L\text{-}i\text{-}1,\, i)\,, \tag{4.64}$$

$$\vdots$$

$$i \to i+2 \qquad \textit{with probability} \quad P_{Inc}(2,\, i)\,, \tag{4.65}$$

$$i \to i+1 \qquad \textit{with probability} \quad P_{Inc}(1,\, i)\,, \tag{4.66}$$

$$i \to i \qquad \textit{with probability} \quad P_{NoCh}(i)\,, \tag{4.67}$$

and $$i \to i-1 \qquad \textit{with probability} \quad P_S(i). \tag{4.68}$$

Using these event probabilities, and considering the passage of one time unit, we obtain:

$$u_i = 1 \qquad \textit{by definition} \qquad \textit{for} \qquad i = 0, \tag{4.69}$$

and $$u_i = \sum_{x=1}^{L-i}\left(P_{Inc}(x,i)\cdot u_{i+x}\right) + P_S(i)\cdot u_{i-1} + P_{NoCh}(i)\cdot u_i \quad \textit{for} \quad 1 \le i < L\,. \tag{4.70}$$

Rearranging (4.70), we get a general equation in terms of the Markov chain birth and death rates,

$$u_i = \sum_{x=1}^{L-i}\left(\Pi_{up}(x,i)\cdot u_{i+x}\right) + \Pi_{dn}(i)\cdot u_{i-1} \qquad \textit{for} \qquad 1 \le i < L\,. \tag{4.71}$$

Note that the birth and death rates now must take into account the added dimension of "how many states are being transited", when a state increase event happens. That is, $\Pi_{up}(x,i) = \dfrac{P_{Inc}(x,i)}{\sum_{x=1}^{L-i} P_{Inc}(x,i) + P_S(i)}$ and

$\Pi_{dn}(i) = \dfrac{P_S(i)}{\sum_{x=1}^{L-i} P_{Inc}(x,i) + P_S(i)}$ represent the conditional probabilities of the state transitions $i \to i+x$ and $i \to i-1$ respectively, given that a transition out of state i does occur. It is also to be observed that $\Pi_{dn}(i) + \sum_{x=1}^{L-i} \Pi_{up}(x,i) = 1$, since by definition of the BIN model, these L - i + 1 transitions are the only ones possible out of state i.

Note that we have a special case when $i=L$, since the following relations hold:

$$\Pi_{up}(x, L) = 0, \text{ for all } x \quad (\text{since } P_{Inc}(x, L) = 0 \text{ for all } x) \tag{4.72}$$

$$\Pi_{dn}(L) = 1, \quad (\text{since } P_{Inc}(x, L) = 0 \text{ for all } x) \tag{4.73}$$

Using the (4.73) result, and noting that when $i=L$, the general recursive equation (4.71) has no "increase by x" component, we get

$$\begin{aligned} u_L &= \Pi_{dn}(L) \cdot u_{L-1} \\ \therefore u_L &= u_{L-1} \end{aligned} \tag{4.74}$$

Next, for $i=L$-1 we use the fact that $\Pi_{dn}(L-1) + \Pi_{up}(1, L-1) = 1$, and back-substitute (4.74) into equation (4.71), to obtain,

$$\begin{aligned} u_{L-1} &= \Pi_{up}(1, L-1) \cdot u_L + \Pi_{dn}(L-1) \cdot u_{L-2} \\ u_{L-1} - \Pi_{up}(1, L-1) \cdot u_L &= \Pi_{dn}(L-1) \cdot u_{L-2} \\ u_{L-1} \cdot (1 - \Pi_{up}(1, L-1)) &= \Pi_{dn}(L-1) \cdot u_{L-2} \\ u_{L-1} \cdot \Pi_{dn}(L-1) &= \Pi_{dn}(L-1) \cdot u_{L-2} \\ \therefore u_{L-1} &= u_{L-2} \end{aligned} \tag{4.75}$$

Continuing this process, we derive a general expression relating the probability of absorption from initial state i to that from initial state i-1,

$$\begin{aligned} u_i &= \sum_{x=1}^{L-i} \Pi_{up}(x,i) \cdot u_{i+x} + \Pi_{dn}(i) \cdot u_{i-1} \\ u_i \cdot \left(\sum_{x=1}^{L-i} \Pi_{up}(x,i) + \Pi_{dn}(i) \right) &= \sum_{x=1}^{L-i} \Pi_{up}(x,i) \cdot u_{i+x} + \Pi_{dn}(i) \cdot u_{i-1} \\ \Pi_{dn}(i) \cdot (u_i - u_{i-1}) &= \sum_{x=1}^{L-i} \left(\Pi_{up}(x,i) \cdot (u_{i+x} - u_i) \right) \\ \Pi_{dn}(i) \cdot (u_i - u_{i-1}) &= 0 \\ \therefore u_i &= u_{i-1}, \qquad \textit{for} \quad 1 \le i \le L \end{aligned} \tag{4.76}$$

Equation (4.76) combined with the knowledge that absorption from state 0, into state 0 is a certainty by definition (i.e. that $u_0 = 1$), gives, as for the MSV model,

$$u_L = u_{L-1} = u_{L-2} = \ldots\ldots = u_2 = u_1 = u_0 = 1 \tag{4.77}$$

which explicitly tells us that for this finite state-space, limited-station Binary arrival model, *the probability of absorption into state 0 is always 1, regardless of the initial state i.* This absorption analysis, identical for both the BIN and MSV models, highlights the similarity between the two, and is intuitive given that the MSV model is simply a special case of the more general BIN model arrival model.

4.4.2 Calculating the Average Length of the CRI T_c

4.4.2.1 Numerical Recursion

As for the MSV model, the mean time to absorption from an initial state of N outstanding requests may be calculated by iterating t in the state transition equations (4.59) - (4.62) to get $P(0, t)$ and then using equation (4.42) to give $T_C(L, N, p, Perr, \lambda)$.

Consistent with Section 4.3.2.1, the same termination condition is used in obtaining T_C: that of $t \cdot P(0, t) < \varepsilon$ with $\varepsilon = 10^{-9}$. Use of equation (4.42) in this BIN model scenario will once again only give a valid approximation to the true value of T_c if the $\{L, N, p, Perr, \lambda\}$ parameter values are such that the PPA into state 0 is very close to 1, as stated explicitly in equation (4.43).

4.4.2.2 Analytical Solution

Following the same methodology and notation of [KARL 75] (and Section 4.3.2.2), given that absorption is certain from any initial state i, let ω_i denote the mean time taken to be absorbed into state 0, from the initial state i. We obtain a recursion formula for ω_i by enumerating the only possible states after the first timeslot has passed. Using the definitions of (4.63) - (4.68) for the event probabilities, we get:

$$\omega_i = 0 \quad \textit{by definition} \quad \textit{for} \quad i = 0, \tag{4.78}$$

$$\text{and} \quad \omega_i = 1 + \sum_{x=1}^{L-i} \left(P_{Inc}(x,i) \cdot \omega_{i+x} \right) + P_S(i) \cdot \omega_{i-1} + P_{NoCh}(i) \cdot \omega_i \qquad \textit{for} \qquad 1 \le i < L\,. \tag{4.79}$$

Rearranging (4.79), we obtain an equation of the form used in [KARL 75]:

$$\omega_i = \left(\frac{1}{\sum_{x=1}^{L-i} P_{Inc}(x,i) + P_S(i)} \right) + \sum_{x=1}^{L-i} \left(\Pi_{up}(x,i) \cdot \omega_{i+x} \right) + \Pi_{dn}(i) \cdot \omega_{i-1} \qquad \textit{for} \qquad 1 \le i < L\,. \tag{4.80}$$

with $\Pi_{up}(x,i)$ and $\Pi_{dn}(i)$ as defined in 4.4.1. Using the earlier result $\Pi_{dn}(L)=1$, and noting that when $i=L$, the general recursive equation (4.80) has no "increase by x" components, we get

$$\begin{aligned}\omega_L &= \frac{1}{P_S(L)} + \Pi_{dn}(L)\cdot\omega_{L-1} \\ &= \frac{1}{P_S(L)} + 1\cdot\omega_{L-1} \\ \therefore \xi_{L,L-1} &= \omega_L - \omega_{L-1} = \frac{1}{P_S(L)}\end{aligned} \tag{4.81}$$

Further back-substitution of (4.81) into equation (4.80) for $i=L$-1, yields

$$\begin{aligned}\omega_{L-1} &= \left(\frac{1}{P_{Inc}(1,L-1)+P_S(L-1)}\right) + \Pi_{up}(1,L-1)\cdot\omega_L + \Pi_{dn}(L-1)\cdot\omega_{L-2} \\ \omega_{L-1}\cdot(\Pi_{up}(1,L-1)+\Pi_{dn}(L-1)) &= \left(\frac{1}{P_{Inc}(1,L-1)+P_S(L-1)}\right) + \Pi_{up}(1,L-1)\cdot\omega_L + \Pi_{dn}(L-1)\cdot\omega_{L-2} \\ \Pi_{dn}(L-1)\cdot(\omega_{L-1}-\omega_{L-2}) &= \left(\frac{\Pi_{dn}(L-1)}{P_S(L-1)}\right) + \Pi_{up}(1,L-1)\cdot(\omega_L-\omega_{L-1}) \\ (\omega_{L-1}-\omega_{L-2}) &= \left(\frac{1}{P_S(L-1)}\right) + \frac{P_{Inc}(1,L-1)}{P_S(L-1)}\cdot(\omega_L-\omega_{L-1}) \\ \therefore \xi_{L-1,L-2} &= \left(\frac{1}{P_S(L-1)}\right) + \frac{P_{Inc}(1,L-1)}{P_S(L-1)}\cdot\xi_{L,L-1}\end{aligned} \tag{4.82}$$

Continuing this process, we derive a general expression for the mean time to first make the transition $i \to i-1$,

$$\begin{aligned}\omega_l &= \left(\frac{1}{\sum_{x=1}^{L-i} P_{Inc}(x,i) + P_S(i)}\right) + \sum_{x=1}^{L-i}\left(\Pi_{up}(x,i)\cdot\omega_{l+x}\right) + \Pi_{dn}(i)\cdot\omega_{l-1} \\ \omega_l\cdot\left(\sum_{x=1}^{L-i}\Pi_{up}(x,i)+\Pi_{dn}(i)\right) &= \left(\frac{1}{\sum_{x=1}^{L-i} P_{Inc}(x,i) + P_S(i)}\right) + \sum_{x=1}^{L-i}\left(\Pi_{up}(x,i)\cdot\omega_{l+x}\right) + \Pi_{dn}(i)\cdot\omega_{l-1} \\ \Pi_{dn}(i)\cdot\left(\omega_l-\omega_{l-1}\right) &= \left(\frac{1}{\sum_{x=1}^{L-i} P_{Inc}(x,i) + P_S(i)}\right) + \sum_{x=1}^{L-i}\left(\Pi_{up}(x,i)\cdot(\omega_{l+x}-\omega_l)\right) \\ \omega_l-\omega_{l-1} &= \left(\frac{1}{P_S(i)}\right) + \sum_{x=1}^{L-i}\left(\frac{P_{Inc}(x,i)}{P_S(i)}\cdot(\omega_{l+x}-\omega_l)\right) \\ \therefore \xi_{i,i-1} &= \left(\frac{1}{P_S(i)}\right) + \sum_{x=1}^{L-i}\left(\frac{P_{Inc}(x,i)}{P_S(i)}\cdot\sum_{k=1}^{x}\xi_{i+k,i+k-1}\right), \qquad for\ \ 1\le i\le L\end{aligned} \tag{4.83}$$

The mean time to absorption into state 0, from initial state $i=N$, is again the sum of the component steps $\xi_{i,i-1}$ as given earlier by (4.50). Contrasting the equation for the average step duration of this BIN model we have just derived, (4.83), to that of the MSV model, (4.49), yields some interesting observations. Namely, in the MSV model, the fact that during any one timeslot only one arrival is possible, leads to the average duration of the i^{th} step (down to state i-1) depending on the previous $(i+1)^{st}$ step only. However, the ability of a BIN system in state i to record anywhere between 1 and L-i arrivals in a single timeslot means that the i^{th} step (down to state i-1) depends directly on all of the previous steps: $i+1 \to i$, $i+2 \to i+1$, ... up to $L \to L-1$. This general relationship would seem to make the divergence of the i^{th} step $\xi_{i,i-1}$ more likely for $i \to 1$, over a broader range of the system's parameter values $\{L, N, p, Perr, \lambda\}$, and hence make it more difficult for the BIN model to converge to a PPA of 1 and a "finite" T_C for practical purposes, as compared to an MSV model. However, this recursive-step effect very strongly depends on the $P_{Inc}(x, i)^{BIN}$ and $P_S(i)^{BIN}$ values; and, as we shall show in the results section, the differing characteristics of the BIN and MSV models cause non-linearities and changes of sign in the differences $(P_{Inc}(i)^{MSV} - \sum_{x=1}^{L-i} P_{Inc}(x,i)^{BIN})$ and $(P_S(i)^{MSV} - P_S(i)^{BIN})$. It will be illustrated in the ensuing section, that the magnitude and sign of these differences determine the relative strengths of each model's state-dependent "*upwards pull*". It is this upwards pull, defined as a ratio between the probability of state increase to the probability of state decrease, which has the biggest impact on the PPA and T_C behaviour of the two models.

4.4.3 Obtaining the Critical Load, L_{crit}

In order to obtain the critical load, L_{crit}, for the Binomial model, we use equations of the same form as those that were presented in Section 4.3.3. That is,

$$L_{crit} = \frac{N + E[arr\_during\_T_C]^{BIN\_model}}{T_C} \tag{4.84}$$

with the mean number of background arrivals during T_C for the Binomial model, given by

$$\begin{aligned} E[arr\_during\_T_C]^{BIN\_model} &= \sum_{t=1}^{\infty} P(0,t) \cdot \left(\sum_{\tau=1}^{t} E[arr,\tau]^{BIN\_model} \right) \\ &= \sum_{t=1}^{\infty} P(0,t) \cdot \left(\sum_{\tau=1}^{t} \left(\sum_{j=1}^{L} P(j,\tau) \cdot E[arr,j]^{BIN\_model} \right) \right) \\ &= \sum_{t=1}^{\infty} P(0,t) \cdot \left(\sum_{\tau=1}^{t} \left(\sum_{j=1}^{L} P(j,\tau) \cdot \frac{L-j}{L} \cdot \lambda \right) \right) \end{aligned} \tag{4.85}$$

Equation (4.85) ends up identical to its counterpart from Section 4.3.3 (equation (4.52)), since the mean arrival rate in state *j* is the same for both the MSV and BIN models. That is,

$$E[arr, j]^{MSV\_model} = \frac{L - j}{L} \cdot \lambda \,, \tag{4.86}$$

$$E[arr, j]^{BIN\_model} = (L - j) \cdot \left(\frac{\lambda}{L} \right), \tag{4.87}$$

by virtue of our parameter matching method which was discussed earlier.

4.5 Signalling Capacity Allocation Schemes

We now consider three different schemes to manage and access CMS signalling slots. Up until now, in the analysis of Sections 4.1 through to 4.4, we have assumed no special arrangements for the use of the signalling channel: only one CMS was assumed per upstream data slot, and all stations were free to use it without any group-based or cycle-based restrictions. As we shall see from the numerical results presented in Chapter 6, introducing some of these TDM-related concepts can have significant benefits with regards to the attainable critical load.

Let *M* be the number of CMS's associated with each upstream data slot (ATM cell). The first scheme, where all stations may access any of the *M* slots, is termed *Full CMS sharing with multiple CMS's per data slot (abbreviated as FCS)*, with a critical load given by,

$$L_{crit} = \frac{MN}{T_C(FCS, N, p, P_{err})} \tag{4.88}$$

The critical load, L_{crit}, is obtained as the ratio of the amount of work arriving in a batch, namely *N*, and the average time in *upstream slots* (defined as super-slots containing one data slot and the *M* associated CMS's) during which the contention resolution of these *N* requests is completed, which is equal to T_C / M upstream slots. Note that L_{crit} is a signalling channel quantity measured in requests per upstream slot - it is considered critical in the sense that if the mean interarrival times of size *N* request batches is lower than T_C, the signalling queue becomes unstable. Under such circumstances, the signalling queue would continuously grow and ultimately undergo congestion collapse. In essence, all equations giving L_{crit} define the stability criterion for the HFC system's virtual signalling queue.

In considering all of these CMS management and allocation schemes, we assume an underlying extreme-case model of a batch of *N* single cell messages arriving all at once and generating *N* simultaneous requests. It is interesting to note that the worst performance is exhibited when the data load and signalling load are equal (single-cell messages); situations where the message size is large (i.e. message size $\rightarrow N$ cells) would perform better because a given data load would be generating a signalling load which is only a fraction of itself. For example, if the batch of *N* cells was being generated by the arrival of *N/2* messages

of size 2 cells each, the signalling (CMS) load would only stand at *N/2* requests per upstream slot. It is intuitive that a best case scenario for signalling performance in the context of the model just presented, would be the contention-free case of an arrival of a single *N*-cell message.

Under the second scheme, the *N* stations are subdivided into *M* groups, so that each group must access a different CMS associated with the data slot. In this fashion the effective load accessing any given CMS is reduced to *N/M*. Implementation-wise, the station counts passing CMS's and is allowed to access every M^{th} CMS, which occurs once per data slot since we have *M* CMS's for every one data slot. We term this scheme *Cyclic CMS sharing with multiple CMS's per data slot (or CCS_M for short)*, with a critical load given by,

$$L_{crit} = M \cdot \left\{ \frac{\frac{N}{M}}{T_C(CCS\_M, \frac{N}{M}, p, P_{err})} \right\} = \frac{N}{T_C(CCS\_M, \frac{N}{M}, p, P_{err})} \tag{4.89}$$

Lastly, the third scheme, similar to the multi-CMS Cyclic sharing, is its *single-CMS* variant (and is termed C*CS_S*). Using only one CMS per data slot, if we divide the *N* stations into *k* groups, whereby each group may only access every k^{th} CMS, we obtain a critical load of,

$$L_{crit} = \frac{N}{k \cdot T_C(CCS\_S, \frac{N}{k}, p, P_{err})} \tag{4.90}$$

Recalling the definition of the average CRI duration, T_C, there are four common factors which affect the critical load: (i) the CMS scheme used together with the number of CMS's per timeslot, (ii) the batch size, *N*, (iii) the p-persistence probability *p*, and (iv) the probability of CMS error, P_{err}. Note that if single-CMS Cyclic Sharing is used, there is an additional fifth factor - the number of separate contention resolution groups, *k*. Both variants of Cyclic CMS sharing, (CCS_M and CCS_S), may in some sense be viewed as the *p*-persistence algorithm's non-adaptive version of the collision-slot grouping concept from *START-n* [BISD 96a].

5. Fair Centralised Priority Reservation (F-CPR): A Candidate IEEE 802.14 Protocol

5.1 Background

The primary contribution of Section 3.3.5 has been to provide an overall understanding of the basic properties of, and issues associated with, HFC MAC protocols and concrete examples thereof, with a particular insight into how best to design (i) a MAC protocol and (ii) associated Head-End scheduling algorithm, which efficiently and cost-effectively support multiple QoS traffic classes. Other than the generic HFC MAC protocol design guidelines which were referred to in Section 3.3.5.2, the following "implementation specific" conclusions may be drawn from the overall study in Section 3.3.5. Note that the conclusions also include the Head-End scheduler functions:

- The protocol should be centralised, and be capable of allowing DPD while the Head-End scheme should be information based (a variant of EDD) and capable of rescheduling; this combination would allow the bandwidth allocation process to be delayed until "the last possible moment", with the associated efficiencies.

- The capability of immediate access should not be considered a critical requirement, and can thus be left out, unless it comes as an *in-built* part of the protocol as in the case of DQRAP [XU 93]. In general, the immediate access capability has tended to give better QoS than required for certain traffic classes, thus wasting the implementation effort.

- The piggybacked DMS unit is an imperative, given that the shared transmission medium has a very long propagation delay, and is prone to collisions.

- The MAC frame structure should flexibly handle any mix of the supported traffic classes, using the concept of dynamically adjustable intra-frame boundaries explained in [SALA 95].

- Supporting a deterministic access delay bound on a contention-based shared medium is a very difficult and expensive undertaking; moreover, by using statistical bounds it is possible to obtain a QoS equivalent to deterministic bounds, at the price of not meeting the bound in the rarest of cases.

The CPR protocol [LIMB 95], [SALA 96a] was designed with the above set of principles in mind. The Fair-CPR or F-CPR protocol is our particular implementation of the generic CPR protocol. Although we call our version of the protocol Fair CPR, we do not claim that F-CPR is fairer than CPR; rather, it is a specific implementation of CPR where certain fairness related alternatives (for handling the existing and new users at the Head-End bandwidth manager), not discussed in [SALA 96a], have been specifically implemented. We now focus our studies on F-CPR, which, as detailed earlier, bears a very high degree of similarity to the December 1996 draft IEEE 802.14 standard (as well as IBM's formally submitted MLAP protocol).

5.2 Protocol Description

The aim of our description of F-CPR is to provide a full and detailed specification of a MAC protocol based on CPR; but, unlike CPR, to also include detailed guidelines about all aspects of operation, from the user stations to the Head-End controller's basic bandwidth management functionality (the more complex multi- and single-priority scheduling is considered by the IEEE 802.14 WG as an "add-on" on top of any MAC protocol). In essence, the CPR is a framework, while F-CPR is a comprehensive MAC level specification that the access network design engineer can physically implement (on top of an appropriately specified physical layer).

It is important to note that the fairness feature gives F-CPR its name. When the Head-End receives each upstream data slot, it has to make two key decisions. The first is the order in which the CMS and DMS requests are to be served; the second is exactly what to do if only one part of a request can be scheduled contiguously (i.e. if the user station has to be told to break its message into two or more bursts). In the specification of F-CPR, we explicitly address these two issues, in such a way that some degree of fairness is introduced when serving users generating CMS requests on one hand and users generating DMS requests on the other. Given that DMS requests are generated by users who have already got control of the channel, we should try to avoid giving such requests further preferential treatment. The CPR specification does not explicitly address the two above-mentioned Head-End decision processes, and so any generic implementation may be assumed, potentially at the detriment of global fairness (which may be viewed as equal access to channel bandwidth AND Head-End service capacity, for all users). In Chapter 6 we shall show the difference in performance between one such generic implementation of CPR and our F-CPR approach.

Since F-CPR is a specific implementation of the CPR protocol, most of the features are identical to pure CPR. In the upstream direction, stations transmit information in fixed length data slots (like [SALA 96a], this work is based on one ATM cell as the entire data slot payload), to which is appended a minislot field, used for reserving further bandwidth (only used when a station is sending the final data slot of its allocation, and has one or more further messages enqueued in its local message buffer). This fixed length data slot plus reservation minislot is only accessible to a particular station, which has been "scheduled" by system control to write in it. The presence of directional taps (as discussed in Section 3.3.5.1) means that reading is impossible in the upstream direction.

Between these fixed length data and reservation composite units we have the much smaller "minislots", which are used purely for contention, and as one such minislot passes from the furthest end of the cable to the Head-End, any station may attempt to write its request details (identification, priority and number of requested cells) in it. It is clear that a contention resolution scheme needs to be implemented here.

The Head-End reads a data slot / contention minislot pair (for all intents and purposes simultaneously) in a periodic manner, and then schedules the requests which have just arrived, if any. It may happen that these fields are empty (no requests), or contain valid request information (either a successful contention or a

further-allocation request or both). Note also that a "garbled" contention field may arrive in the event of two or more stations writing in it. This is considered a collision, and nothing is done by the Head-End. Finally, as will be explained in Section 5.2.3, the bandwidth management algorithm performed by the Head-End is fair, in the sense that an implicit priority is given to the most recently arrived contention request, over the most recently arrived data reservation request. The next step is the process of scheduling, which is performed by the use of two lookahead buffers within the Head-End (called the Data Image (DI) and Grant Image (GI) buffers), where upstream data and downstream grant slots are marked for a given station. The scheduling can thus be done with a view to the future, and it is completed at the appropriately marked time (obtained from the GI buffer) by sending an addressed grant minislot field (which exists as an independent part of a downstream data slot) in a “just-in-time” manner to the relevant station. This removes the need for much intelligence at the station, and allows the station to transmit data as soon as it receives the grant.

In the downstream direction, data slots will be addressed to certain stations, yet they will contain two independent minislot fields within them, which are addressed to be read by different stations. In this way, all stations snoop the medium and only read a given field when they recognise their address. When a station receives an acknowledgement message, it knows that it was successful in its transmission via the contention upstream channel. What follows then is the reception of a grant allocation message, which will alert the station to begin its transmission in the immediately following upstream data slot, for the specified duration of slots.

5.2.1 Slot Structure

Figure 5.1 illustrates the various data slot fields associated with the upstream and downstream paths respectively, and introduces the standard acronyms used when describing any variant of the CPR protocol (CMS, DMS and ACK/GR messages). Note that in this implementation of F-CPR, we have chosen only a single CMS minislot, for the same reasons outlined in [SALA 96a] (minimal benefit of having more than one CMS for the significant associated complexity cost).

5.2.1.1 Data Slots

The data structure within the Upstream/Downstream frames depends on whatever higher layer protocols are employed. In the case of the F-CPR implementation, we have chosen an ATM cell format (53 Bytes, including 5 Bytes of header information per cell).

5.2.1.2 Contention Minislot (CMS) Field

These request messages may be “contested” by any stations along the line, as the frame passes them. The stations are aware of the time, in slots, that an Acknowledgement (ACK) message takes to arrive back, in the case of success. If this does not happen, a collision has occurred and re-transmission occurs. As was discussed in Section 3.3.5.2.4, the p-persistence CRA is one of the very few algorithms which may be

effortlessly implemented in a configuration which supports a DPD environment. As a result, p-persistence is the chosen CRA in the F-CPR protocol, just as it is in the original CPR version.

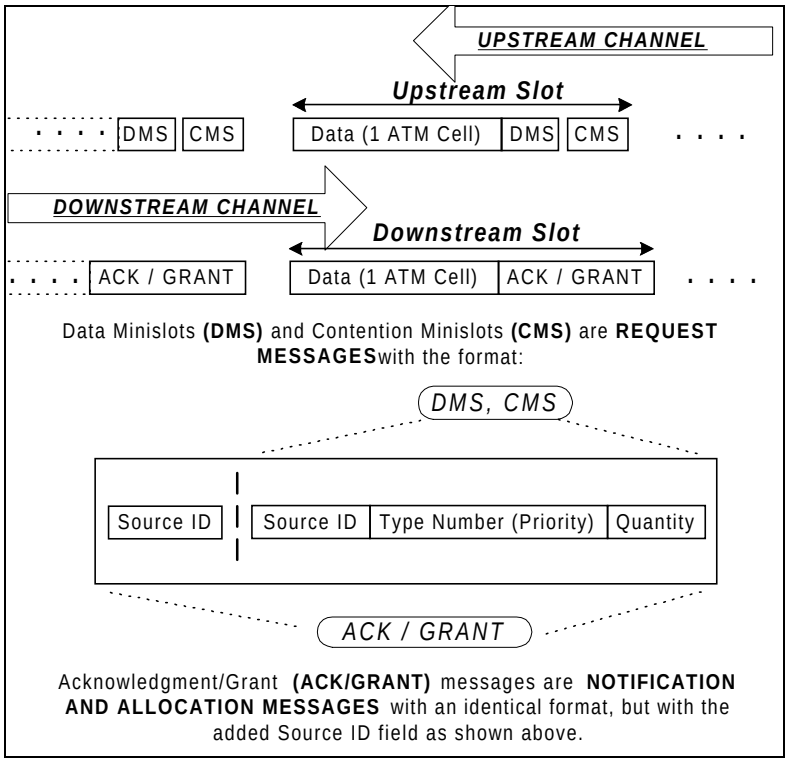


Figure 5.1: The Slot Structure Used in the F-CPR Protocol

5.2.1.3 Data Minislot (DMS) Field

Once a station comes to the end of its current message, it may write a further request (if its message queue is non-empty) into this slot. ONLY this station may perform this write - it is therefore considered strictly a **reserved field.**

5.2.1.4 Ack/Grant Minislots

An ACK message signifies a particular station's success in the contention process, and tells it to shortly expect a GRANT message. The GRANT may even be in the same downstream data slot as the ACK, and it tells the station how many contiguous upstream data slots have been allocated to it, starting from the next upstream data slot.

5.2.1.5 "Typical" Field Bit-Sizes:

Below, we outline the expected sizes of the individual fields shown in Figure 5.1:

- *Source ID field* - between 11 and 16 bits (up to 65,535 addressable stations).
- *Type Number* - between 1 and 3 bits (possible support for ATM traffic classes).
- *Quantity* - in this implementation 5 bits (no more than 30 Cells per single msg.).

Therefore:

- *CMS or DMS minislot field* size ~ 3 bytes.
- *ACK/GR minislot field* size ~ 5 bytes.

Giving:

- *Upstream Slot* size ~ 53 + 3 + 3 bytes.
- *Downstream Slot* size ~ 53 + 5 bytes.

Given that the CMS and DMS fields are considered as a total 6 byte signalling overhead on top of the upstream data cell (which is 53 bytes), we can state that the upstream Data Link Layer Efficiency (*DLLE*), is 90%. The definition of DLLE appears in [LIN 95], and represents the portion of the data link capacity which is available to carry user data, excluding physical layer considerations such as preamble and guard band components (see below). Similarly the ACK/GRANT field is a 5 byte notification overhead in the downstream, giving a downstream DLLE of 91%.

However, to each of these slot sizes, we need to add the framing overhead in bytes associated with guard bands and synchronisation preamble codes, if we want to find the overall protocol efficiency (*OPE*). The term OPE is used to denote the proportion of the channel's bandwidth actually available to the user data. Therefore, this efficiency figure is calculated as a simple ratio of the size of the user data in bytes to the total number of bytes, including all signalling and framing overheads, required for the transmission of this user data (whether it be down- or upstream). The framing overheads are a function of the physical layer, and from [LIMB 95, MAC1 96] we see that a total of 8 bytes is required for *each contiguous transmission slot* (defined as a separately writable number of bits). Examples of a contiguous transmission slot are (i) the entire Downstream Slot (since only one station, the Head-End, can ever write into it), (ii) the Data plus DMS portion of the Upstream Slot, and (iii) the CMS portion of the Upstream Slot. Note that (ii) and (iii) have to be separate contiguous transmission slots, since two stations must be able to separately and at different times write data into them.

This framing aspect of the CPR and F-CPR protocols means that each Downstream Slot, with a size of 58 bytes, requires just one set of preamble and guard bands resulting in an 8 byte framing overhead and a downstream OPE = 80%. On the other hand, every Upstream Slot has a size of 59 bytes, with the CMS as a separate contiguous transmission slot requiring its own guard bands and preamble, and thus two sets of preamble and guard bands are required, resulting in a 16 byte framing overhead and hence a reduced 71% upstream OPE value.

5.2.2 Description of Station Actions

The transitions which a station makes during normal operation, and the states it resides in are illustrated in Figure 5.2.

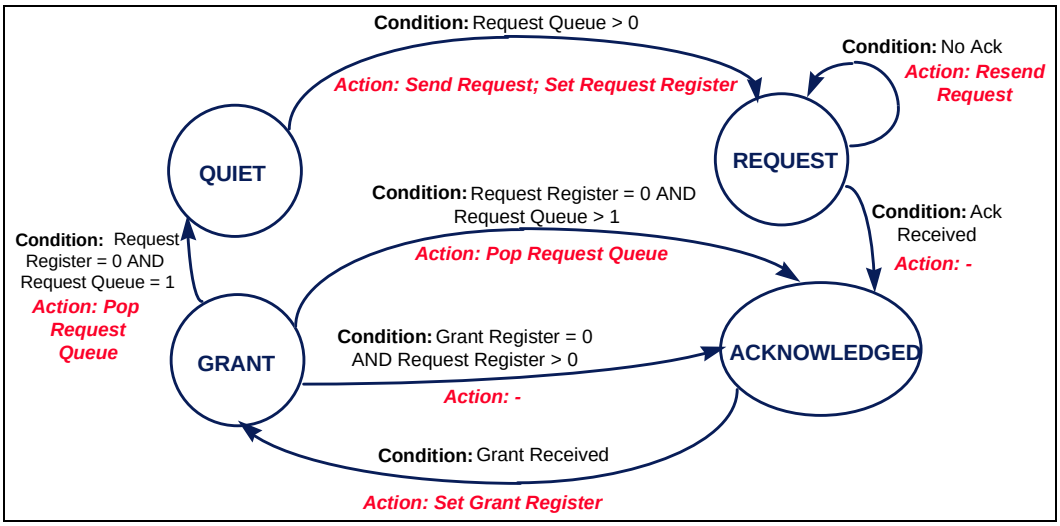


Figure 5.2: The Station Protocol State Transition Diagram

Upon a message's arrival, it is appended to the end of the request queue (FIFO). Its priority (type) is determined at this point, as is the number of slots required to fully carry the message contents. Note that in F-CPR, one slot corresponds to one ATM cell, and in this implementation we "capped" the maximum request size to 30 ATM cells per single message, in order to capture no more than a maximum size Ethernet packet (~1500 Bytes). The reasoning behind this decision had two main arguments: (i) Most of today's Cable Modem manufacturers have opted for a 10BaseT Ethernet-style interface between the user's PC and modem unit, and (ii) this protocol is to be subjected to tests in our work under a real Ethernet traffic trace. Many different higher layer protocols (IP is a very good example) may "sit on top of" Ethernet, and ultimately be passed into the Cable Modem unit.

It is thus likely that the ubiquity of the Ethernet network card in today's PC world, coupled with its proven performance characteristics and excellent speed (10Mbit/s interface) will mean that a maximum message size, (in cells/slots), close to the one we have chosen, will not even have to be enforced by the station. It will rather be a normal by-product of using the Ethernet interface between Cable Modem and PC.

Key Points regarding Protocol Operation:

- A station is aware (i.e. it is previously synchronised) of the time, in slots, when the ACK message should arrive if it has been successful in the contention phase (i.e. if it has successfully made the transition from the *Request* state to the *Acknowledged* state).

- When a station in the *Acknowledged* state snoops the bus and finds its own address (Source ID), it begins to transmit the specified no. of cells in the very next upstream timeslot, after setting its Grant Register.

- Transmission of the entire message unit is only complete when both the Request AND Grant Registers have been counted down to 0. As Figure 5.2 shows, this means that sometimes less slots

will be granted than were requested, and hence the station will have to segment its message unit into two or more "bursts" of cells.

- If the Head-End is able to immediately grant the requested number of slots, then the Ack and Grant will arrive in the same downstream timeslot. The implication of this is that a station needs to be able to *immediately* read the GRANT field, having just read its own address in the ACK field. This situation equates to two state transitions occurring *{Request -> Acknowledged -> Grant}* within one timeslot.

5.2.3 Description of Head-End Actions

The transitions which the Head-End makes during normal operation, and the states it resides in are illustrated in Figure 5.3 (overleaf). The transition *{Ack Sent -> QUIET}* with the condition that the DMS is not empty, denotes the relatively rare case where both the contention (CMS) and reserved (DMS) fields are found by the Head-End to contain valid requests.

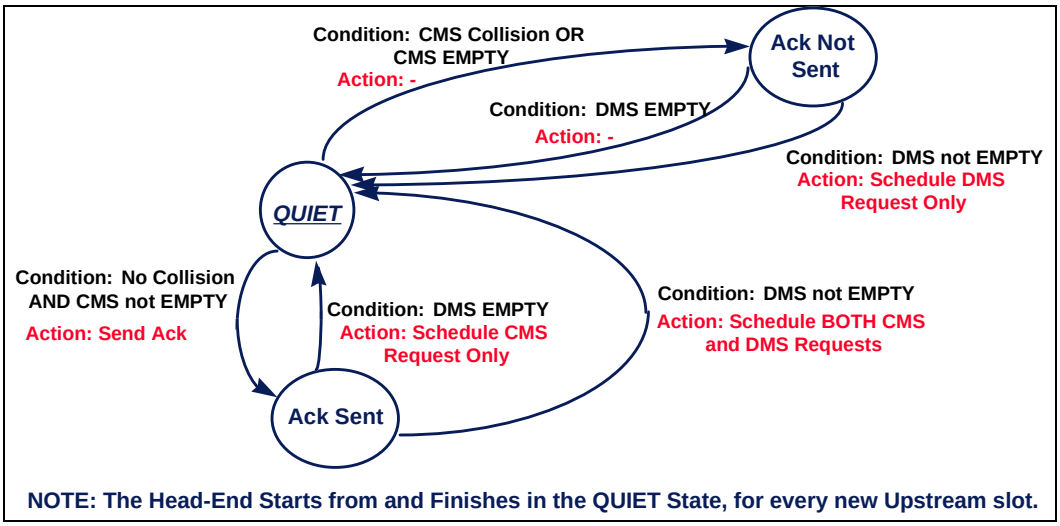


Figure 5.3: The Head-End Protocol State Transition Diagram

The stated fairness of the F-CPR protocol relates to this particular situation - prior to actually scheduling a request into the lookahead buffers, the protocol tries to appropriately allocate upstream transmission bandwidth to the requests belonging to the contention and reserved fields, respectively:

(1) The station *contending for the upstream channel* (CMS field) has been quiet, unlike the station which is sending the request in the reserved (DMS) field. As a result, the CMS request is the first to undergo the initial Head-End scheduling iteration - one type of fairness is introduced here, we label it *Type 1 fairness*.

(2) If the Head-End has NOT been able to completely schedule the requested number of contiguous slots for the CMS request, it then attempts to completely schedule the (DMS) *reserved request*.

(3) This alternating process of scheduling a part of one and then the other request introduces a certain amount of overall fairness, which we label *Type 2 fairness*.

The process which we term *scheduling* is a process performed by means of examining the lookahead memory (DI and GI buffers) within the Head-End, that contains an image of future upstream data slots and downstream grant slots, in which previously occupied slots have been already marked. When a request is to be scheduled, the Head-End first scans the DI buffer for the first free upstream data slot and calculates the number of contiguous free slots available, following this first one. As will be discussed in more detail later on in this section, it is not however always possible for the Head-End to allocate this first available free upstream data slot to a station, since the associated grant slot may not be free. If the associated grant slot IS free, on the other hand, the Head-End then performs the schedule by marking its DI and GI buffers, and at the appropriate time sends an appropriately addressed grant minislot field (which exists as an independent part of a downstream data slot), to the relevant station.

While points (1) through (3), together with the presented state transition diagram form the slightly modified First-In-First-Out basis of the Head-End scheduling algorithm, a number of different “add-on” components need to be implemented depending on the characteristics of the HFC system. This will all depend on the equipment manufacturers, since, as highlighted previously, the Head-End scheduling algorithm is going to be omitted from the IEEE 802.14 standard specification. One of the add-on components will be a priority scheduling scheme, designed to cater for multi-priority systems capable of supporting multimedia traffic with ATM-like QoS requirements, as discussed in Section 3.3.5.4. The performance of two such schemes, and benchmarking against an ideal single server queue with many priorities, is the topic of separate work, presented later, in Section 6.3.

However, another important add-on component is the algorithm which copes with the distorted request arrival times and grant channel congestion, in a contention-based system with many stations at different propagation delays (PDs) along the shared distribution medium. One such algorithm, together with a discussion of its relevance to the work presented here, is mentioned in Section 5.2.3.1 below.

5.2.3.1 Scheduling to Support a System with DPD

When the Head-End schedules any request, it does so by keeping a DI buffer with an image of future data slots in which granted upstream slots have been marked, as well as the GI buffer, with an image of future grant slots in which allocated downstream grant slots have been marked. The Head-End must also be aware of each station’s PD, so that it can successfully schedule the requests. Any system that times the grants so that the station can transmit immediately upon their reception (in the next upstream slot), suffers from congestion in the grant channel if stations are located at different PDs along the medium. A typical situation illustrating grant channel congestion is one where it may not be possible to schedule the first available upstream data slot, since another station (with a different PD) has had its grant already scheduled to be sent by the Head-End at that same time. In such a case, the issuing of the grant would need to be

delayed until a free pairing of data - grant slots was found in the upstream and downstream channels respectively (see Figure 5.4 for an illustration of this phenomenon).

[SALA 96b] discusses the nature of the modifications required in adapting well-known ATM scheduling algorithms for use in an HFC system, within the Head-End scheduler. The same paper explains that an ATM switch scheduler operates with the actual data in real time, and is thus very close to the *ideal scheduler* - one which has a global view of all the information to be scheduled and can thus make optimal decisions. HFC schedulers are physically constrained to operating on the basis of requests rather than a true real-time awareness of the actual data. Therefore an ideal HFC scheduler would need to have unlimited-capacity reservation and grant channels, so that any station requests could be immediately communicated to the Head-End collision-free, and any grants would be delivered at the optimal time. It would also be required that a station can support any number of outstanding requests. Real HFC systems are unfortunately quite limited in all respects, and these limitations introduce the following problems:

- **Traffic Distortion** -
  a limited reservation channel introduces collisions in the request transmissions, and hence causes a distortion between the traffic specified by a station during a connection admission control (CAC) phase, and the actual traffic arriving at the scheduler.

- **Service Delay Distortion due to Outstanding Requests -**
  the limitation of the number of outstanding requests per station introduces a variation in message service delay. In many HFC protocols (including both F-CPR and CPR), a new request cannot be sent until the previous one has been served. Therefore, consecutive requests will arrive at the Head-End in intervals with duration equal to or greater than the time taken to complete servicing the previous request. The service delay is variable and depends on the amount of resources available in the network (both the shared medium and Head-End processor). As system load increases and queues start to build up, this highly variable service delay causes the traffic to undergo service delay distortion.

- **Service Delay Distortion due to a Limited Grant Channel -**
  the part of service delay distortion which is attributable to the limited grant channel, resides in the fact that it is not possible to simultaneously send more than one grant message per time slot, causing a further and variable grant-congestion related delay. Note that a grant channel capacity equal to DPD (in slots) would theoretically be enough to eliminate any grant channel congestion. An alternative remedy to grant channel congestion is for all stations to be virtually moved to the end of the line by using a VDB, as discussed in Section 3.3.5.2.3. Most HFC protocols trying to eliminate grant channel congestion focus on the latter of these two solutions, but, as [SALA 96b] points out, *"the move of stations to the end of the line will be worthwhile only if the increase in average RTD is smaller than the delay... (which would otherwise be incurred due to)... grant channel congestion."*

The same two Rate-Controlled approaches to dealing with Traffic and Service Delay distortion used in switched networks, may be applied in HFC networks [ZHAN 95]. The first approach is work-conserving, and requires modelling of the specific HFC distortion introduced to the traffic characteristics specified by

the CAC scheme. Thus, the distortion model used in an ATM network's CAC would need to be augmented by the HFC model. The second approach is non work-conserving, as it involves a regulator at the Head-End (c.f. policing mechanism at an ATM switch) which avoids distortion by holding the traffic until it follows the description given in the CAC (i.e. hence it is labelled as non work-conserving).

The models and results presented in [SALA 96b] are based on the latter of the two approaches, whereby a non work-conserving Head-End is modelled as a combination of *scheduler* and *grant generator*. It is stated that the function of the scheduler is to decide which request goes next, while the grant generator is responsible for finding a valid grant-data slot pairing. The ATM scheduling discipline chosen in [SALA 96b] for embedding into an HFC system is Self Clocked Fair Queueing (SCFQ) [GOLE 94]. The results of the SCFQ testing illustrate that:

- In an HFC system where all stations are at the same location, or have been virtually moved to the end of the line, a switch scheduler (SCFQ) may be directly applied with **no added functionality**, and the same capabilities in guaranteeing QoS as in the switching environment may be maintained.

- In an HFC system with a limited-capacity grant channel and the stations at varying PDs along the line, **added functionality must be added to the switch scheduler**, in order to guarantee collision-free upstream data transmission efficiently. This added functionality comes in the form of a grant generator module, which converts ordinary grants into *valid grants* - permissions to use free and correctly matched data-grant slot pairs.

- The particular scenarios which were tested focused on the trade-off between the benefit and cost of virtually moving stations to the end of the shared line: the benefit being elimination of grant channel congestion, but the cost being a longer average round trip delay. In the tested scenarios it was found that the cost outweighed the benefit, and that the overall mean access delay (i.e. the mean waiting time for a cell, from its arrival at a station's queue to its transmission) for the variable PD HFC system was smaller than that of the system with fixed PD.

As mentioned above, grant channel congestion was modelled in the [SALA 96b] paper by the Head-End component module called the *grant generator*. In a variable PD system it was found that the delay introduced within the grant generator stems from the impossibility of scheduling the optimal grant, not from avoiding upstream data collisions. On the other hands, since in a fixed PD system the optimal grant is always available (it is never required to send two or more grants simultaneously) there is zero delay within the grant generator. This, however, is offset by the longer average round trip delay for such a system. It is instructive to illustrate how grant channel congestion affects upstream data delay in a very simple example system with two stations each trying to transmit a four cell message and configurations as shown in Figure 5.4. The times at which the transmission of both four-cell messages is complete are circled in bold ellipses for each of the three different system configurations. Note also that the station ID numbers are written into the relevant time slots occupied either by their data cells (Upstream Channel) or grant notifications (Downstream Channel).

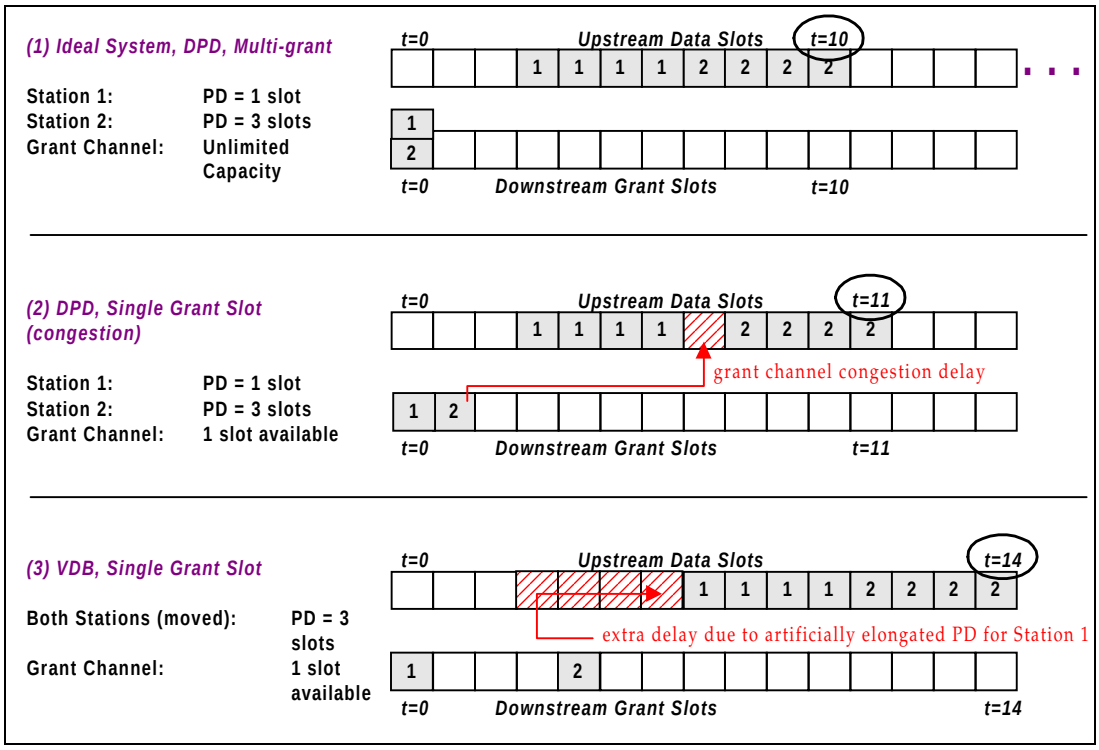


Figure 5.4: The Effect of Grant Channel Congestion on Upstream Delay

It is evident that system configuration 1 is ideal, with a minimum completion time of only 10 slots. The worst result is achieved when both stations are "padded" with virtual delay buffers, and thus moved to the end of the line, as shown in configuration 3 where the transmission ends at slot 14. As shown in the figure, the entire extra four slots of delay in configuration 3 (as compared to configuration 1) come as a result of the artificially elongated PD for station 1. On the other hand, we see in configuration 2 that the penalty of congestion in the limited grant channel is only one slot, with a total transmission time of 11 slots. Clearly, this example scenario reinforces the finding of [SALA 96b], that even though the benefit of VDB is to eliminate grant channel congestion, in many (if not most) system configurations the penalty of a significantly higher round trip delay has a more detrimental effect and thus causes worse overall system delay performance.

Figure 5.4 also highlights the fact that the grant channel congestion penalty is usually not very significant. In [SALA 96a] it was stated that even in the *"... most extreme cases (of system configuration) we found this factor to have a negligible effect"*. In fact, most HFC configurations will be far from extreme (PD and upstream transmission speed -wise), especially with the proliferation of fibre serving areas as HFC networks grow, where even very fast upstream systems (~ 8-10 Mbit/s) are physically spread over geographically smaller and smaller areas, until almost all stations lie within 1 or 2 slots PD from the Head-End. Theoretically the grant channel congestion problem can be all but eradicated by using more than one grant minislot per upstream data slot. However, the simple example above and the formal conclusions in [SALA 96a, SALA 96b] suggest that the added complexity and reduced framing efficiency of such a solution is not at all warranted, given the negligible effect of the grant channel congestion problem in all HFC systems studied thus far.

5.2.3.2 Fragmentation Scheduling in A System with DPD

A final issue related to efficient scheduling of upstream data slots is the special feature of the CPR and F-CPR protocols which needs to be mentioned in the context of HFC systems with DPD. Optimally, and especially at low loads and for fixed propagation delay systems, each station will receive its original number of requested data slots in one contiguous chunk. However, in DPD systems, and particularly where the cabled tree branch topology is extremely long, it is not uncommon for the Head-End to have to break up and schedule requests into many smaller sized "piecewise chunks", so as to avoid upstream collisions while maintaining maximal line utilisation. Hence, both the CPR and F-CPR protocols have an embedded *fragmentation scheduling* feature, as evident from the stations' state transition diagram in Figure 5.2. Namely, a station may request a given number of upstream slots, but in the interests of optimal system wide efficiency, the Head-End may decide to allocate these upstream slots through two or more grant messages, forcing the station to break up its message into multiple bursts.

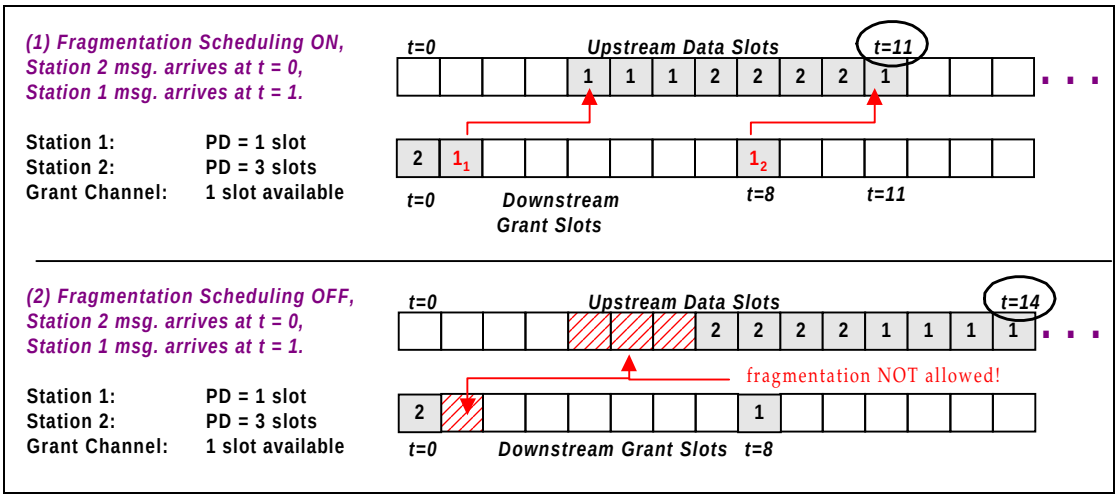


Figure 5.5: The Benefit of Fragmentation Scheduling

A simple example system of two stations each trying to send a four cell message, is presented in Figure 5.5, with the fragmentation feature switched on and off, in order to contrast the resulting upstream line efficiency. As in the previous example, the station ID numbers are written into the relevant time slots occupied either by the stations' data cells (Upstream Channel) or grant notifications (Downstream Channel). It is clear that the system with the fragmentation feature enabled enjoys a substantially shorter overall transmit time, completing the whole transmission at time slot 11 instead of time slot 14.

5.2.4 Specific Implementation Options

It is important to outline exactly the novel features of our particular implementation of the pure CPR protocol described in [SALA 96a, SALA 96b]. First and foremost, by F-CPR we refer to the entire superset of both station and Head-End functionalities, meaning that we look at the full "end to end" protocol suite. Taking a holistic approach to system performance, we spend an equal amount of time looking at how the Head-End bandwidth management framework (but *not* the specific higher-layer

scheduling scheme) should be designed, as we do in validating the proposed CPR station design. Secondly, in line with the relevant industry trends and most likely evolution path for the cable data modem technology, we make or use the following assumptions about some system-wide parameters and features (most of which were not made/used in the earlier studies of the original CPR [LIMB 95, SALA 96b, SALA 96a]):

- Explicit fairness (two types of fairness have been defined in Section 5.2.3) in the Head-End's pre-scheduling bandwidth management Algorithm (and hence the name F-CPR).

- Given that it is assumed we are working with speeds and distances where long propagation delay lines are unlikely, and also that the grant channel congestion problem has been satisfactorily researched in [SALA 96b], F-CPR only implements a simple, slightly modified version of FIFO scheduling, for single priority systems.

- For multi-priority systems, F-CPR may implement a variety of priority scheduling disciplines, but the one we chose to implement and test extensively is *Just-in-Time Exit Timestamp (JET).*

- The Data Information Field has a fixed payload unit length, which is large enough to encapsulate one ATM cell of 53 Bytes.

- The maximum number of slots which can ever be asked for in a single station request physically cannot exceed 30 (which corresponds to the maximum-size Ethernet packet, as explained in Section 5.2.2).

5.2.5 Methodology for Supporting Multiple Priorities of Traffic

In Section 3.3.5.4 we discussed the need at the network service provider side for supporting many users with different quality of service (QoS) requirements. The expected need for telcos to be able to seamlessly offer customised services to users of the Internet, to people wanting toll-quality voice telephony, or to business users of video conferencing, while maintaining the guaranteed QoS levels, forces the 802.14 MAC candidate protocols to introduce support for multiple traffic priorities and also forces the equipment (i.e. Head-End and user set-top unit) manufacturers to implement methods of handling multiple priority traffic (*priority and scheduling schemes*), which may be proprietary or in the public domain.

In this section, we describe two such multi-priority schemes, implemented *as an added layer of intelligence on top of the F-CPR MAC protocol*. Sections 5.2.5.1 and 5.2.5.2 below provide a clear functional boundary between the priority and scheduling **components** of the two multi-priority schemes which are under investigation: *Just-in-Time Exit Timestamp (JET)* and *Scheduling Advance (SA).* In Chapter 6, we will present performance evaluation results for these two multi-priority schemes, using the same real Ethernet measurements which will be used for single priority protocol testing. Since the original real trace contains unprioritised traffic, it is clear that we need the means for somehow assigning levels to

the priority of traffic emitted by each station. To this end, Section 5.2.5.3 discusses the ways in which we assign priority levels to the unprioritised real trace which we measured and described in Chapter 2.

5.2.5.1 The Priority Component Scheme

This component scheme consists of two parts. The first part is entirely within the functionality of the MAC protocol, in the form of a Type field within a CMS or DMS minislot. This field was not used at all in our earlier investigations of the protocol due to the single priority nature of the traffic, but can be used to denote different priorities of traffic. The integer zero is used for the highest priority traffic, and currently the number of priorities supported is not specified yet. Based on the three most popular types of currently used ATM Transfer Capabilities (CBR, VBR and UBR) we opted to conduct our investigations with three different priorities, with the integer two representing the lowest priority.

The second part of the priority component scheme consists of: (i) the queue entry discipline which is an "add-on" specification not residing within the MAC protocol functionality; and (ii) an associated multi-round queue manipulation framework which is within MAC protocol functionality since it is directly derived from the "fair" bandwidth management feature of F-CPR, given in Sections 5.2 and 5.2.4. Both of the multipriority traffic handling methods under consideration, JET and SA, were designed with this component scheme functionality being common, as shown in Figure 5.6 and described below.

Firstly, the entire priority-scheduling process is initiated each time a new upstream data slot reaches the Head-End, and the process is completed in one or more rounds. The reason for the multi-round nature of the process shown in Figure 5.6, is that the Head-End, having found the first useable free upstream data slot, reserves subsequent slots in the DI buffer until the number of requested has been reached or an already-used slot is encountered. A second round will be necessary if an already-used slot is encountered, and thus some slots still remain to be scheduled. It is therefore conceivable that a single request might result in more than one grant to the sending station. This fragmentation means that the Head-End can "pack" upstream data slots in a very efficient manner.

As shown in Figure 5.6, the queue entry discipline within our priority component scheme dictates that the Head-End either stores any low priority request in the relevant low priority buffer (b-1 buffers exist for b priorities), or immediately schedules any high priority request, with first preference afforded to the CMS request. This preference for processing the CMS instead of the DMS requests first, is not actually part of this priority scheme; rather, it is a derivative of the explicitly "fair" bandwidth management framework implemented in F-CPR, as the reader will recall from Sections 5.2 and 5.2.4. This framework is therefore common to all priority schemes and resides *within* the boundary of the actual F-CPR protocol. Namely, we choose to attempt scheduling the contents of the CMS request slot prior to those of the DMS request slot, in order to be fair to the winner of the contention phase who has not even begun to transmit yet. In addition, if subsequent rounds of scheduling are necessary, and both DMS and CMS requests are still competing, once again the priority will be afforded to the newly successful CMS request (see the k^{th} round illustration within Figure 5.6).

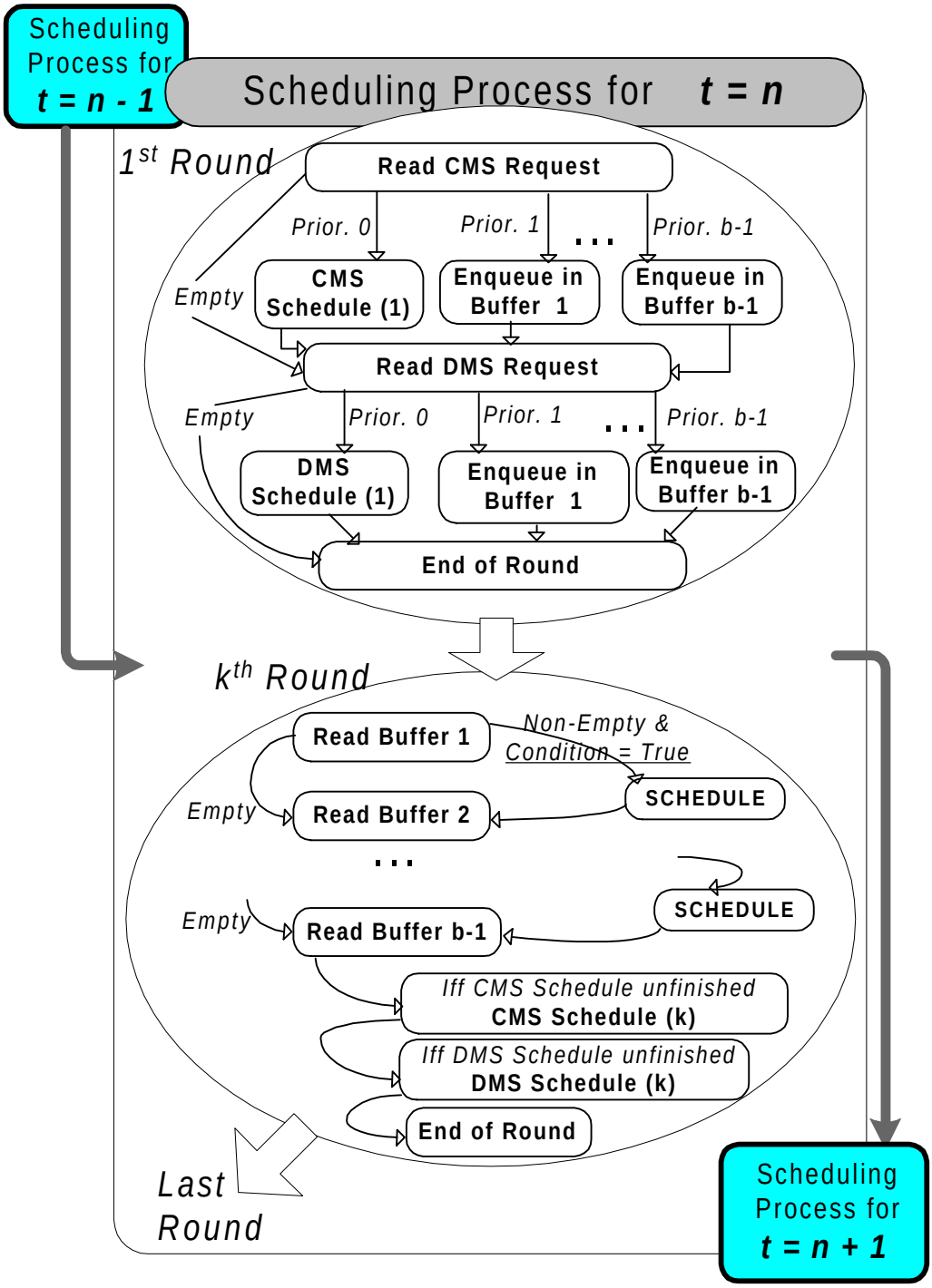


Figure 5.6: Multi-Round Priority Scheme at the Head-End

Turning our attention now to the actual priority component scheme (residing on top of F-CPR's explicitly fair bandwidth management framework), we note that it has been designed so that *b-1* buffers exist for *b* priorities, and only the highest priority requests (i.e. priority 0) are NOT buffered in the first round of scheduling (topmost illustration in Figure 5.6). All other priorities are first sent to auxiliary holding queues, (which employ the FIFO discipline). In all subsequent rounds of scheduling (e.g. k^{th} as highlighted in the figure), the Head-End cycles through these holding queues (descending from the second highest priority down). If the holding queues are not empty, and if the relevant scheduling condition is satisfied, the Head-End attempts to schedule the requests. Otherwise no action is performed, either because the request is as yet not allowed to exit the holding queue or the queue is empty.

5.2.5.2 The Scheduling Component Scheme

The type of this component scheme used, is that which clearly differentiates the JET method from the SA method.

5.2.5.2.1 JET Scheduling Scheme

This scheme uses a scheduling condition which assigns a *just-in-time exit timestamp* when storing lower priority requests in their separate holding queues. The JET is calculated as the first usable future timeslot (such that **both** the relevant grant and upstream data slots are free, when the DI and GI buffers are scanned) minus the propagation delay from the Head-End to the station in question. This is illustrated in Figure 5.7 (overleaf). It is always desirable to design a scheme which enables fine control of the relative levels of priority between different traffic types. Although we shall not delve into detailed study of this aspect of the schemes, we now show that this can be achieved easily in the JET scheme.

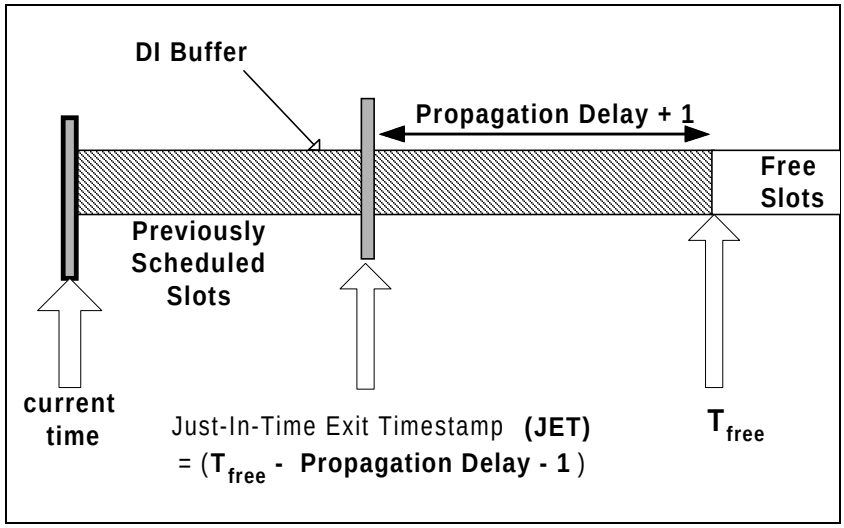


Figure 5.7: Snapshot of Data Image Buffer for the JET Scheduling Scheme

The **earliest** time of queue exit is the current time, corresponding to immediate scheduling and virtually no priority differentiation (other than order of queue exit). On the other hand, as has just been illustrated, the **latest** queue exit time that can be assigned to a request is the JET. The difference (JET - current time) will always be zero or positive, depending on the state of the DI buffer. Therefore, by introducing a percentage quantity known as the Delay Factor ($0\% \leq DF \leq 100\%$), we can issue exit timestamps (ETs) such that,

$$ET = \textit{current time} + DF \cdot (JET - \text{current time}) \qquad (5.1)$$

This results in a control mechanism whereby minimal priority differentiation is achieved by setting DF = 0%, such that ET = current time; and, maximal priority differentiation is achieved by setting DF = 100% such that ET = JET. The subject of our scheme comparisons in this work is the case with DF = 100%, since maximal priority differentiation is the most frequently sought outcome, and the ultimate objective is the best possible utilisation and delay performance for the highest priority traffic (comparable to that of a

multi-priority single server queue). Figure 5.7 reflects this, and from this point on we strictly assume that ET always equals JET (i.e. the lower priority requests are maximally delayed). Note that with the JET method an added functionality on the part of the Head-End is required: when inserting requests into the lower priority queues it must calculate the JET, and upon expiration of the JET (if required) it must recalculate a new JET (explained below).

The request at the head of each low priority holding queue has the exit condition:

exit and be scheduled

only if ① current time = JET

and if ② the grant and data image buffers are still both free at the slots *at (current time+ propagation) for GI and (current slot + propagation + 1) for DI* respectively.

If the current time gets to the JET, and the head-of-queue request's target usable slot is no longer free, then the JET is re-calculated as the next usable future upstream slot, and the request remains enqueued. Naturally, the queue ordering remains unchanged and in this way requests of the **same priority** are strictly served on a FIFO principle.

A final statement remains to be made, about the **pre-emptive** nature of the queueing discipline intrinsic to this scheduling scheme. As was explained earlier, only the highest priority (type 0) traffic is never enqueued and is scheduled immediately, while the medium and low priority traffic must first enter their own appropriate holding queues, and then be served only after the above mentioned scheduling conditions are met. The definition of pre-emption is the interruption of service of a lower priority request, caused by the later arrival of a higher priority request. It is clear that the low priority requests may never pre-empt any other request priority, because of their position at the bottom of the "priority ladder". However, the medium and high priority requests can be both thought of as having the ability to "interrupt" the scheduling process to all of the priority types below them on the priority ladder. For example, regardless of the JET assigned to a medium and/or low priority request which had arrived many slots ago and queued up for service (and been assigned its "scheduling ticket"), if a high priority request arrives at any time prior to this JET, it can usurp this scheduling ticket. The same logic applies when a medium priority request has the chance to do the same to a low priority request.

It may be confusing to some as to why this is a pre-emptive discipline, when the requests being overtaken by higher priority requests are seemingly never actually interrupted in the middle of *service time*. This is where the definition of *service* plays a very important role. Namely, the service for high priority requests consists solely of the scheduling, and it is a one-step operation completed in one timeslot (maybe multiple rounds, but this is not relevant). On the other hand, the service time for medium and low priority requests commences when they are assigned a JET and enter a queue, and concludes only when they successfully exit the queue and are scheduled, which can be many timeslots later. Therefore, whenever a lower priority request attempts to leave the head of the queue with its valid JET, and is blocked by a higher priority

request being scheduled in the meantime, a pre-emption of service has occurred. The pre-emptive property of the JET scheduling component scheme, as will be shown later on in this section, plays a very significant role in improving the high priority requests' average access delay and utilisation performance of the JET traffic handling method, with respect to that of the SA method.

5.2.5.2.2 Scheduling Advance Scheme [LIMB 95]

This scheduling component scheme, proposed by Sala and Limb in [LIMB 95], revolves around only scheduling lower priority requests for a given number of slots into the future. The scheme uses a scheduling condition which prohibits the Head-End from scheduling a request if the **first** usable future timeslot (i.e. T_{free} in Figure 5.8, overleaf) is found to be more than a certain number of slots into the future. This number is termed the *scheduling advance (SA)*, as shown in Figure 5.8. The figure gives an example where the Head-End allows low-priority traffic to *only be scheduled into the first five possible slots*. Because the first usable slot is found to be the fifth possible slot, the scheduling may go ahead and the request will leave the holding queue.

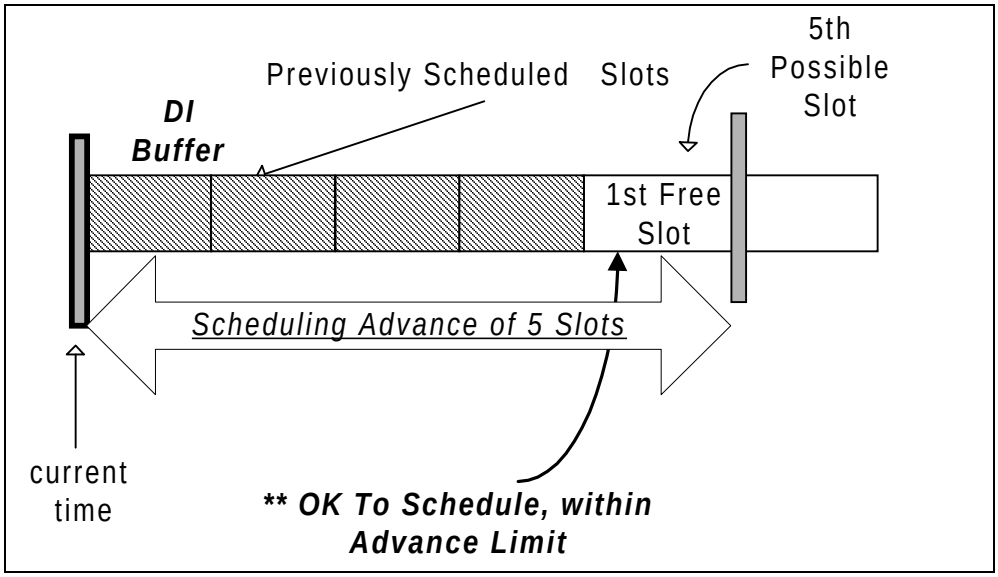


Figure 5.8: Snapshot of the Data Image Buffer for the Scheduling Advance Scheme

The authors of [LIMB 95] state that this choice of priority scheduling scheme was made in order to achieve a degree of control over the relative levels of priority between different traffic types. More priority (less scheduling delays) can be given to the lower priority traffic types by scheduling them more than one slot into the future. That is by *increasing the allowed scheduling advance.* In comparison with the JET method, this would directly correspond to issuing a JET with a smaller DF (i.e. allowing a lower priority request to attempt scheduling earlier than the last possible opportunity). As was explained in the case of the JET scheme, because we are after maximal priority differentiation, we only conducted simulations with SA = 1, which would give the lowest performance for the lower priorities (equivalent to setting DF = 100% for the JET scheme).

5.2.5.3 Choice of Priority Assignment Mechanism

Having determined which priority scheduling schemes will be used to handle traffic of multiple priority levels, the next step to modelling a realistic HFC system is the choice of a priority assignment mechanism. Or, in other words, given the characteristics of the current trace read process, how should the stations generate multi-priority traffic? As will be demonstrated later, this is not a trivial question, since the method of generating multiple-priority traffic has a significant impact on the delay performance of all scheduling schemes, as well as the IM itself.

There are two approaches to the generation of multi-priority traffic, each based on its own assignment mechanism. The first, and comparatively simpler approach to choose is that which has been used to obtain the results presented in [LIMB 95], and which we label Randomly Mixed Priorities (RMP) priority assignment.

The basic concept is illustrated in Figure 5.9. Put simply, any station in the system can generate messages of any priority; the probability density function (of generating a certain priority message) within each station can be arbitrary. The simplest implementation of this is where the distribution is **uniform**, and **equal probabilities of occurrence** are assigned to each of the priority levels. In the system we are studying, there are three levels, 0 through to 2, with 0 being the highest priority. Hence, if we label with M_p the assigned priority of the message just generated at a station, equation (5.2) shows that on average M_p will be assigned with equal frequency to priorities 0, 1 and 2:

$$\Pr(M_p = x) = \frac{1}{3}, \quad x \in \{0,1,2\} \tag{5.2}$$

A comment needs to be made about RMP systems: the priority assignment mechanism partially destroys the intra-station correlation structure of each and every station, on a per-priority basis. That is, the overall correlation of the original arrival stream is still intact if we do not look at priorities, but for each individual priority type it is reduced because any long bursts will on average be diluted equally among all three priorities.

The second approach to multi-priority traffic generation is based on the principle of having as many station subsets, with different sized populations, as there are priority levels in the system. Each subset is associated with a given priority, and a station is assigned permanent membership to such a subset for the duration of the simulation, and can only generate messages of this one kind. This type of system is termed Priority Groups (PG) assignment, and is shown in Figure 5.10 (overleaf). The population of each subset can vary from zero to the total system size, but we have decided to choose the populations so as to equalise the loads coming from each of the three priorities (reason for this given below). Let us label the priority assigned to a message in a station belonging to a priority y subset, with $M_p(y)$; we may then describe the behaviour of a PG system with equation (5.3):

$$\Pr(M_p(y) = x) = \begin{Bmatrix} 0, x \neq y \\ 1, x = y \end{Bmatrix} \tag{5.3}$$

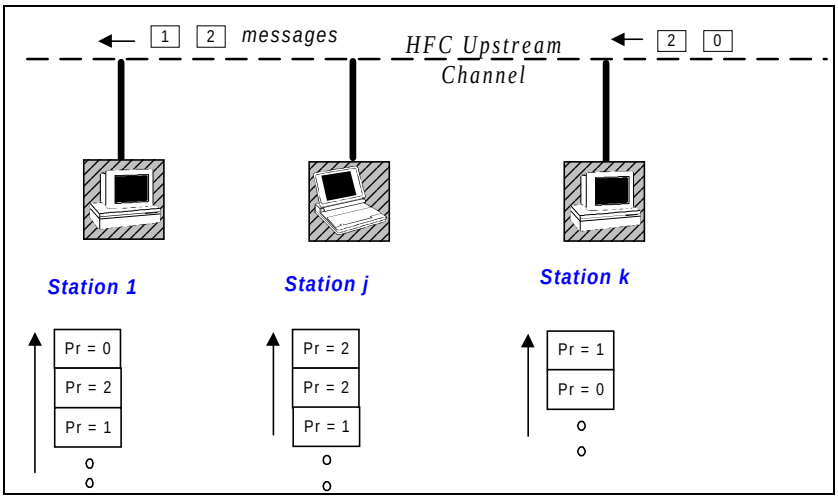


Figure 5.9: Randomly Mixed Priorities (RMP) Assignment System

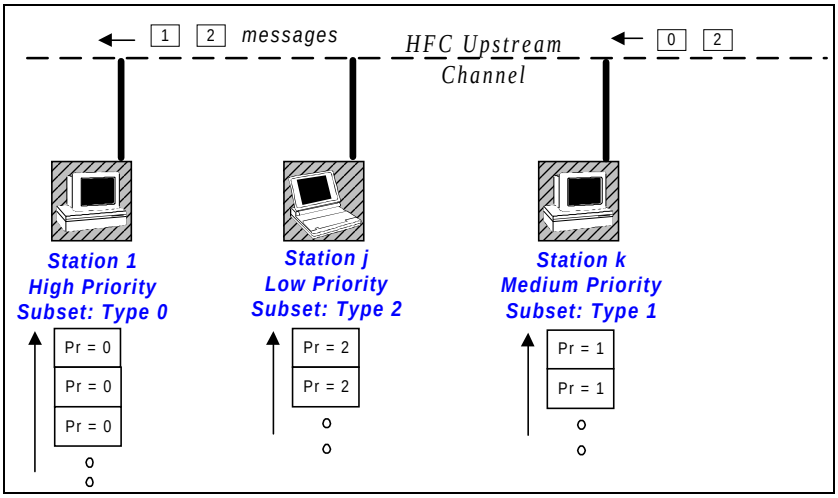


Figure 5.10: Priority Groups (PG) Assignment System

The most likely and realistic scenario of application use among residential and Small Office/Home Office (SOHO) users is expected to result in a system which may be modelled most closely by the PG priority assignment mechanism. That is, we expect that during the timescales which our simulation tests represent (never more than minutes of real time) it is not reasonable to observe all active stations sending each of the three priorities of traffic (in whatever ratio). Rather, there will be variable-size subsets of stations, each of which is using a given type of application and thus requiring a given level of traffic priority. It is improbable that the overall loads (measured in generated cells) associated with each of these subsets will be equal - a time varying distribution is expected. The reason that we have chosen to equate these loads in our tests, is that we wanted to enable a fair comparison between the PG systems and RMP system (which generates equal loads on a per-priority basis).

6. Performance Evaluation of F-CPR

This chapter is devoted to evaluating a comprehensive set of performance criteria for the F-CPR MAC protocol for HFC access networks under diverse conditions. The different conditions are generated from a measured traffic trace which we consider as realistic (refer back to Chapter 2 for details). We include a detailed investigation into conditions leading to signalling channel congestion collapse (deadlock), and evaluate three signalling capacity allocation schemes which we have earlier proposed in order to alleviate the deadlock problem to some extent.

Although all results presented here explicitly refer to a particular HFC access network MAC protocol, the reader should keep in mind that in principle, these results are almost universally applicable to the WATM MAC protocol based on next-generation wireless PCS systems. Section 3.3.6 fully justifies this statement, by describing how similar the WATM and HFC MAC protocols are, both in terms of architecture and design. This is an important paradigm, because it suggests that the conclusions we shall draw from the results in this chapter, *may be considered as conclusions for a multi-service MAC protocol, applicable to both fixed and wireless high speed access networks*.

6.1 Simulation Testing: Method, Traffic Types and Model Parameters

6.1.1 General Method

The simulated F-CPR, CPR and ideal multiplexer systems were programmed using the object-oriented paradigm through a (Visual) C++ platform. The term *slot* (or *timeslot*) was used to represent the time taken to transmit one cell. Our simulations were loaded by two types of traffic: (i) two real traffic traces and (ii) artificial memoryless traffic based on a Bernoulli-Geometric arrival model, with parameters matched to those of the real traffic from the traces. In this way we could both test the protocol under real traffic trace loading, and compare the performance of the protocol when loaded by synthetic memoryless traffic versus its performance when loaded by real traffic. The advantage in doing the comparison by using synthetic traffic from the same Bernoulli-Geometric memoryless traffic model first proposed by Sala and Limb in [SALA 96a], is twofold:

- We can explore the validity and performance of the model proposed by Sala and Limb.
- We can validate our results.

6.1.2 Using Measured and Model-generated Traffic

In order to test the F-CPR protocol with realistic traffic, we use our measured Ethernet trace files. Recall that the details of the measurement set-up, controlling the replay speed of the trace, as well as all relevant statistics, have been presented in detail in Section 2.6.

On the other hand, testing the F-CPR protocol with model-generated traffic, began with the use of the Bernoulli-Geometric traffic arrival model, as used in the simulation testing of the CPR protocol in [SALA 96a]. This model results in a memoryless (in terms of autocorrelation structure) and *independent and identically distributed (i.i.d.)* arrival process. It assumes a set of Bernoulli trials, one for each message arrival, in order to determine the length of messages (Bernoulli because only two values are permitted). The interarrival times are Geometrically distributed. Note that the time-discrete nature of the shared medium is the reason for the subtle difference in nomenclature; that is, if we were considering a continuous time system, the arrival process would be considered to be Bernoulli with Exponentially distributed interarrival times. In either system type however, the overall process may be intuitively thought of as a special type of Poisson process, where an arrival event happens, but then a further "roll of the dice" determines one of the two possible message sizes, for that event.

More specifically, message lengths are i.i.d. and can take two values: 1 and *i*. The ratio between the number of messages of length 1 and that of length *i* is *j*. That is, every message is of size 1 with probability *j/(j+1)* and is of size *i* with probability *1/(j+1)*. The interarrival times of messages are Geometrically distributed with mean *q*. To compare between the performance of the protocols and ideal multiplexer when loaded by the memoryless i.i.d. traffic (as in [SALA 96a]) with that when loaded by real traffic, we went on to fit the parameters *q*, *i*, and *j*.

The message length parameters we have used in the implementation of the memoryless i.i.d. process are *i*=30 and *j*=4.846. These are found to provide the best fit between the peak and mean message sizes of the memoryless traffic model and our LAN traffic trace, *"kp176.dat"*. Note that we have three quantities to potentially fit - mean, peak and variance, and yet are constrained by the model's interdependencies to only treating two parameters as independent. Thus, a decision had to be made about which two of the three quantities were going to be fitted, and which was going to suffer a mismatch as a result. It was opted to fit the mean and peak because this gives the desired worst case scenario in terms of message size distribution, by **maximising the variance**. We then end up with a Bernoulli-type model which will, in a memoryless fashion, assign an arriving message to one of two sizes, in cells - 1 or 30.

In this way, the resulting message size variance of the memoryless model was about twice that of the real traffic's message variance, ($\mathrm{Var}[Msg.\ size]_{\mathrm{model}} = 119.254$ with $\mathrm{Var}[Msg.\ size]_{\mathrm{trace}} = 60.376$). The logic we employed was that since the real trace was expected to be far worse than the model in terms of all performance indicators (due to the correlation structure and uneven load distribution), the model should be made to have the worst-case statistical properties (i.e. highest message size variance possible) in order to give it the best chance of emulating the overall performance of the real trace, when used as input to either the F-CPR protocol or an ideal multiplexer.

The third model parameter *q*, has been omitted from all of the above discussion regarding fitting, because it is a simulation run-time variable, just like the *replay speed* in the case of the real traffic. Both of these parameters serve the same purpose, in providing different levels of offered load. However, they do not

need to be exactly matched to yield identical load points, since we are mainly interested in constructing and comparing throughput and delay curves rather than individual load points.

6.1.3 Performance Indicators: Definitions

We shall use the following definitions from [LIN 95]:

- *Offered Load* - The amount of user data, as a percentage of the usable channel capacity. For example, if a particular simulation run yielded 500,000 usable upstream data slots, during which a total system-wide figure of 750,000 arriving cells was recorded, then the offered load was $750{,}000 / 500{,}000 = 1.5$ for this simulation run.

- *Average Access Delay* - The average cell queueing delay from the time of a cell's arrival at the station, to the time when it is transmitted. As an example, if a three cell message waits for five time slots upon arrival, for transmission to begin, then the following may be observed: the 1st cell of the message has experienced a five time slot access delay; the 2nd cell has experienced an access delay of six time slots, due to the transmit time of its predecessor; the 3rd cell has experienced an access delay of seven time slots. Therefore, in this situation, the average access delay would be the average of the three figures: six time slots.

- *Utilisation* - The percentage of all available data time slots used in the upstream channel. Although a percentage of 100% would correspond to every upstream data slot being utilised, the overall channel usage would be less than this figure because of the overhead associated with having the contention and reservation minislots, as well as the essential guard band and preamble bits (see Chapter 5). Usually, the durations of these overheads depend on the transmission rate and other specific manufacturer implementation details (how many preamble bits, number of CMS and DMS minislots in each upstream slot etc.). For example, in Section 5.2.1.5 it will be shown that over 10% of the upstream channel is not accessible to user data in the {one CMS, one DMS} protocol implementation, even without taking into account the preamble and guard band wastage.

  Therefore, as highlighted in [SALA 96a], in testing the ideal multiplexer and the two protocol implementations, it is much more broadly useful to focus on the utilisation of that part of the channel actually usable by upstream user data, while ignoring the overheads. Following this approach then enables one to directly calculate the actual data rate in upstream bits per second from a raw data figure, without being preoccupied with the data rate wasted on implementation specific details.

- *Data Link Layer Framing Efficiency* - The portion of the data link capacity which is available to user data (F-CPR Protocol = 90% upstream, 91% downstream), excluding physical layer considerations such as preamble and guard band components (Section 5.2.1.5).

- *Overall Efficiency* - The total percentage of the channel bandwidth which is available to user data (F-CPR Protocol = 71% upstream, 80% downstream), including physical layer considerations such as preamble and guard band components (Section 5.2.1.5).

- *Rate of CMS Collisions* - The ratio of the total number of CMS collisions to the total number of transmitted CMS requests. Unlike a CSMA/CD protocol, where it is the actual data messages which collide and are resent, the F-CPR protocol (and almost every other HFC MAC protocol) uses the notion of signalling collisions. This means that there is a one-to-one correspondence between the number of data messages which arrive at a station's buffer, and the number of data messages which are actually transmitted. As a result, we need an alternative measure for the protocol's transmission efficiency, and so choose to monitor the rate of CMS collisions, which may also be thought of as: the probability that a used CMS minislot is subject to a collision.

- *Rate of DMS Usage* - The ratio of the total number of DMS's used to the total number of transmitted messages. Remembering that a DMS cannot be used by a station unless it has something else to transmit at the conclusion of its current bandwidth allocation, it becomes clear that the rate of DMS usage measures the proportion of all transmitted messages that were sent using a collision-free DMS reservation minislot. An inverse relationship is expected between the rate of CMS collisions and the DMS usage rate, because each message transmitted with the aid of the reservation channel is at least one less potential CMS minislot on the CMS signalling channel (more if collisions and retransmissions occur).

6.1.4 Reference Configuration Concept

When testing any system model which has an abundance of parameters that may affect its performance, a commonly adopted approach is to select a *reference configuration*, manifesting itself as a particular set of parameter values, which exists in an infinite or semi-infinite state space of possible parameter value sets. Once the various performance measures of the system, such as delay or throughput versus load curves, are determined for the reference configuration, each of the parameters identified as important can be varied, in turn (keeping all others constant), in order to explore that particular parameter's effect on system performance.

In line with the testing method outlined in Section 6.1.1, we have used the same reference configuration for the simulation as that in [SALA 96a]:

- Number of Stations = 50;
- Length of Fibre Trunk = 2 slots;
- Length of Coax. = 1 slot;
- Head-End Fixed Processing Delay = 1.4 slots;
- Number of Collision Minislots = 1;
- Number of Data Minislots = 1;
- P-persistence probability = 0.2;
- Bernoulli-Geometric i.i.d. Traffic Model - *i, j* matched to real trace.

Note that both distances and times are measured in the same units - slots. It is important to realise that, although the unit is common, translation back to the relevant physical quantity requires differing interpretation:

(a) ***Time*** is measured in multiples of : the time taken to transmit one ATM cell.
Example: it takes 0.424 ms to transmit a cell (at 1 Mbit/s), so a 1 ms Head-End processing overhead takes 2.358 slots to be completed.

(b) ***Distance*** is measured in multiples of : the distance travelled by the signal on the medium during the time it takes to transmit one cell.
Example: it takes 0.424 ms to transmit a cell (at 1 Mbit/s) and assuming $c' = 2.0 \times 10^8$ ms^{-1}, the distance travelled by the signal on the medium is 84.8 km during this time. So, an end-user who is 100 km away from the Head-End, is effectively 1.179 slots distant.

By translating both time and distance related quantities to a common unit, it becomes possible to easily add up quantities such as propagation delays and Head-End processing delays. More importantly however, the unit of measurement already takes into account attributes such as distance and transmission speed, making comparison much simpler. For example, a user is *one slot from the Head-End* in one scenario. If that same user were to relocate to a residence which was physically only half as far from the Head-End as the former one, and at the same time the network transmission speed doubled, the user would, once again, be only *one slot from the Head-End.* Another benefit of this common-unit approach is that it enables the previously mentioned comparison with the work in [SALA 96a] and [LIMB 95].

6.2 Single Priority Systems

6.2.1 F-CPR Delay and Utilisation Performance

6.2.1.1 Dependence on Traffic Type and System Size

In Figure 6.1 (overleaf) we present utilisation versus load curves for the F-CPR protocol, under real and memoryless traffic for the cases of 10 and 50 active stations. We demonstrate that in all cases examined, even in times of extreme overload, the F-CPR protocol does not suffer from congestion collapse. This is consistent with the intuitive argument that there is always sufficient capacity for the signalling minislot traffic, whether it be using the contention or reservation means. However, as we shall see in Section 6.5, extreme signalling traffic profiles can indeed cause signalling channel deadlock, both from a practical and theoretical point of view.

For both system sizes shown in Figure 6.1, the upstream channel is utilised less when loaded by the real traces, as opposed to when loaded by traffic from the memoryless model. This lowered utilisation for the real traces is worse in the 10 stations case, when the curve of the utilisation begins to fall away from the ideal curve (from 0.85 load onwards).

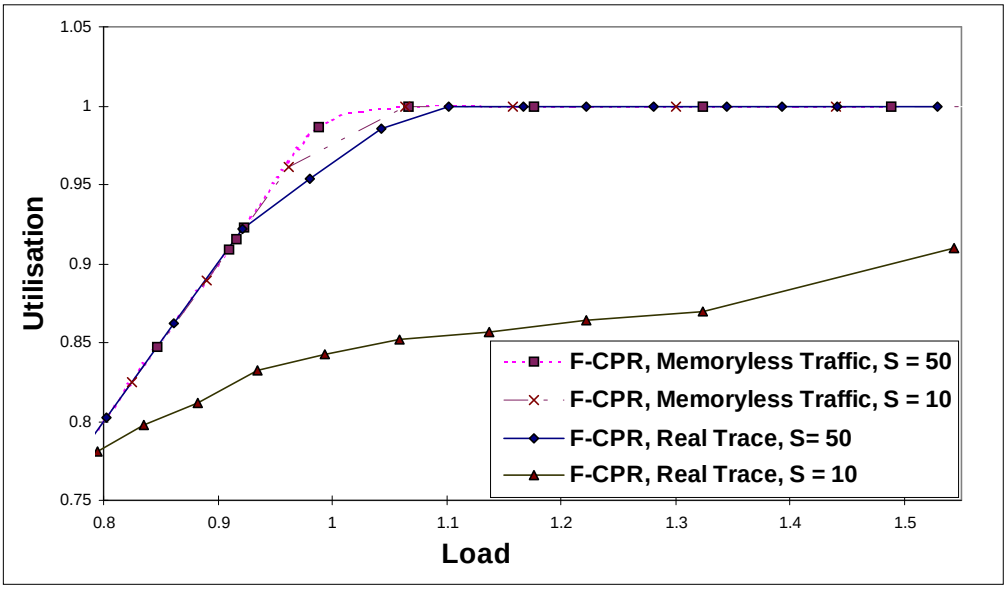


Figure 6.1: Utilisation vs. Offered Load - Real and Memoryless traffic

At 1.0 loading the real trace utilisation is lower by about 0.15 compared with the i.i.d. memoryless traffic case. On the other hand, we do not observe as pronounced a difference in efficiency with 50 stations. Due to the F-CPR's throughput limitations (discussed further in Section 6.2.5) this phenomenon only occurs when a very heavy and correlated load is generated by only one or two stations for long periods of time. The extent of the loss of utilisation efficiency depends on HFC network and station parameters such as coax and fibre distance, upstream transmit speed and average message size. As explained in Section 6.2.5, some rare and extreme system configurations combined with small average message-size traffic from very few stations may exacerbate this problem to more significant levels.

From the point of view of delay performance, the consequences of this throughput limitation are worse. As demonstrated in Figure 6.2 (overleaf), in the case of 10 and 50 active stations each generating a real traffic trace, loads over 0.45 and 0.75 respectively, may lead to unacceptable delay levels. By comparison, the curves based on memoryless traffic are misleading since they give the impression that a load as high as 0.9 will provide acceptable delay levels, regardless of the number of active users. The F-CPR's delay performance in the real trace simulation is highly dependent on the number of active stations. Namely, it significantly improves as the number of stations increases from 10 to 50, and this is consistent with the results of Figure 6.1.

The reason for this improvement with an increased user population comes as a result of two factors: a reduction of the correlation in individual stations' traffic streams and more equal distribution of load among stations. The former is strictly a queueing phenomenon, while the latter aids in the partial elimination of the throughput limitation problem, intrinsic to the F-CPR protocol. Namely, the nature of the recorded trace and the way in which the trace read process functions, means that the correlated load which was originally arriving at only three stations, is now spread to arrive across about 13 stations. Therefore, not only is the correlation partially broken up among more stations, but the probability that only one station will be transmitting for extended periods of time is greatly reduced, causing less RTD-related channel wastage. Also, because the messages are spread over more than one queue now, they are

able to be scheduled in parallel since the Head-End has many simultaneous requests and can efficiently "stuff" the otherwise idle slots on the upstream transmit path. These idle slots are unavoidable if only one station is transmitting all the messages, due to the interaction between RTD and F-CPR's "stop and wait" message transmission regime, as explained in Section 6.2.5.

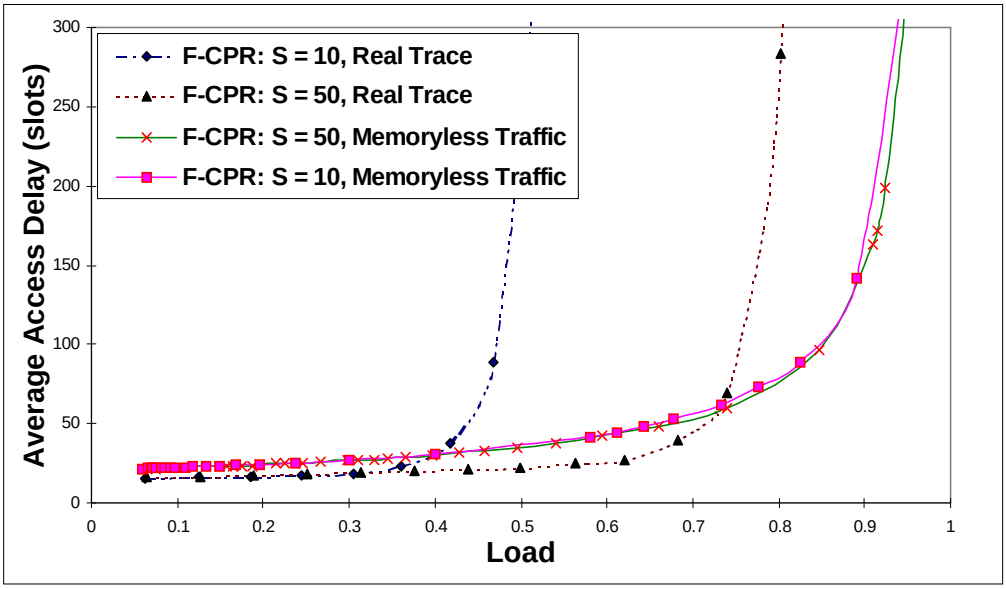


Figure 6.2: Access Delay vs. Offered load under Real and Memoryless traffic

Another interesting feature of Figure 6.2 is that at low loads and regardless of the number of active stations, the average access delay is higher when the protocol is loaded by the memoryless Bernoulli model. This observation may not be intuitive, given the otherwise worse performance of the protocol when loaded by the real trace. The key lies in the worse message size variance of the Bernoulli model. In fact, this variance is double that of the real trace's message size, so it should come as no surprise that, even though the model has the same peak and mean message sizes, its queueing behaviour is worse. Note that this observation is not related at all to the protocol; rather, it is explained purely by well known results from queueing theory [KLEI 76].

6.2.1.2 Impact of Average Message Size on Protocol Performance

The memoryless model average cell access delay curves presented in [LIMB 95], show that the CPR protocol performed identically across all loads from 0 to 1.0, whether there was a small or a large number of stations using the HFC medium. However, Figure 6.3 (overleaf) shows that the memoryless model access delay curves for system sizes of 10, 50 and 100 stations, begin to diverge approximately from a system load of 0.5. This contrasts the appearance of the [LIMB 95] graphs, and the reason lies in the different model parameters used in the two cases. It was stated in [LIMB 95] that an average message size of 4 cells was used. In order to explore the effect of an exceedingly small average message size, our model parameters were set so that either a single or 30 cell message was generated, with an average message size of only 1.58 cells/message.

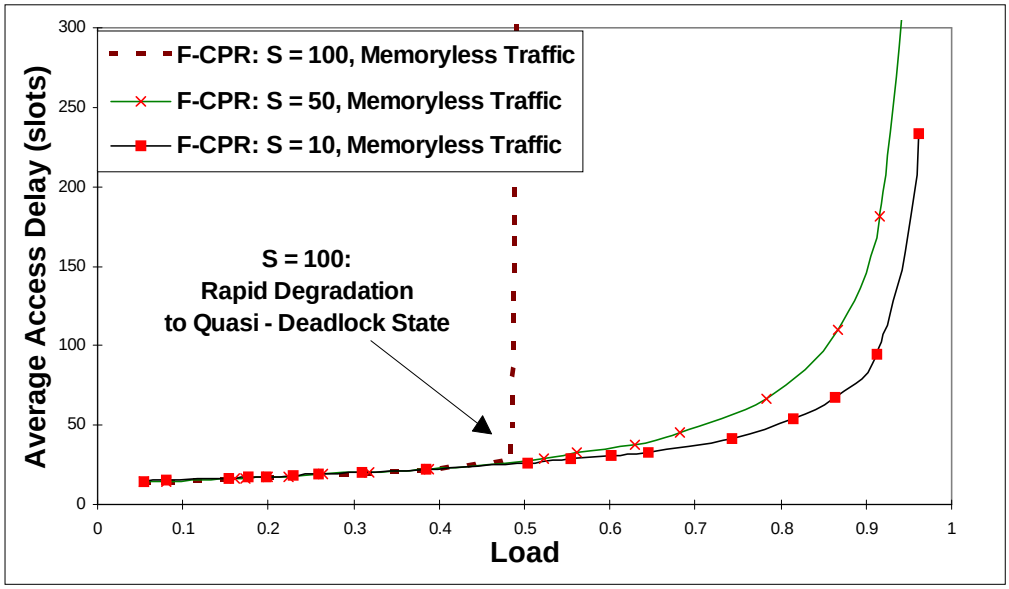


Figure 6.3: Effect of Number of Stations with a small Average Message Size (1.58 cells/messages)

After loads of about 0.5, our mean access delay curves begin to diverge and the performance of the larger systems degrades (100 and 50 user stations). This performance degradation is particularly drastic for the largest of our observed systems (100 stations), where we find an almost vertical asymptote to (practically) infinite delay at a load of ~0.48, indicating a transition to a ***signalling quasi- deadlock state***, which can be explained in the following way. The equally distributed load nature of the memoryless model dictates that the larger systems have more stations generating individually less traffic. But when that traffic has an average message size of 1.58 instead of 4 cells, the potential for collisions is greatly increased because, *in order to generate a given load with smaller messages, more of them have to be sent, and hence more CMS signalling messages are required.* The other characteristic of a model which distributes load equally is that with larger systems there are more stations, each of which gets less of the overall system load; this makes it all the more hard for any given station to have a non-empty queue for any sustained period of time, and hence benefit from the use of the collision-free DMS reservation channel.

The vertical delay curve for 100 stations clearly shows that after a certain load point, if a system has too many stations, the combined detrimental effects of (i) very small average message size (~1.58 cells) and (ii) very low individual station load and hence inability to use DMS reservation minislots in a timely manner, cause a very **un**graceful degradation of delay performance, and lead to a situation where the existing contentions cannot be resolved due to the avalanche of retry- and new-contentions, in a reasonable amount of time (for all intents and purposes - infinity, since most higher layer protocols will time-out after such long access delays).

This is a practically infinite contention resolution interval state, or, “quasi” signalling deadlock state, as referred to previously. The reason that it is not a theoretical deadlock state is that our utilisation versus load curves show that data **still** continues to be sent, and that the channel is still increasingly utilised with increased loading (albeit at a much smaller rate). This suggests that those few stations which were lucky enough to already have control of the channel and non-empty queues, around the time when the deadlock

region was entered, are able to keep clearing their queues. All other stations, which need to compete in order to send their bandwidth requests to the Head-End, repeatedly suffer collisions.

Therefore, in this kind of situation, the presence of F-CPR's contention-free "piggybacked" bandwidth reservation feature keeps data messages moving upstream only until all stations' queues have been emptied. As soon as this happens, the practically infinite contention resolution interval ensures that the system becomes truly deadlocked, if the contention resolution interval is practically infinite. However, in the simulation results we have presented here, there was always more than one station with a non-empty queue, ensuring that at least some portions of the network were able to keep transmitting. Although the delay suffered would probably be intolerable for upper layer (transport) protocols, this scenario has nonetheless highlighted the benefit of the piggybacked reservation message concept. Even with the delay blown out to practically unusable levels, the DMS minislots at the very least postpone a true protocol deadlock and keep the data flowing upstream. At best, DMS minislots help avoid a true protocol deadlock by keeping the message flow alive until a sufficiently long quiet period is encountered to permit the contention resolution algorithm of the CMS signalling channel to clear the backlog of outstanding requests.

As mentioned, from a delay performance point of view, and as seen by a transport protocol such as TCP, the end result is just the same as if a true signalling deadlock and congestion collapse had happened. The *p* parameter of the p-persistence contention resolution algorithm used in the CMS signalling channel would have to be dramatically reduced from its present value of 0.2 (i.e. the contention resolution interval stretched out to accommodate, say, 100 stations) in order to allow F-CPR to better cope with such small-message traffic from so many transmitters. This issue of deadlock will be studied in more detail in Section 6.5.

6.2.1.3 The Danger of Using a Simple Model

A diagram similar to that shown in Figure 6.4, on the next page, was shown in [LIMB 95], with only the average access delay curves for a memoryless model with equal distribution of load among all stations shown, with the following statement being made, *"the (average access delay) results change little if the number of stations increases ..."*. The curves generated by the same equally-distributed-load memoryless model used in [LIMB 95], are shown in Figure 6.4 to tightly corroborate this statement, with very similar numbers of stations used for the test (10, 50 and 100 instead of 20, 50 and 200).

However, in the same graph, we have also included the average access delay curves for three F-CPR systems of the same sizes, and loaded by the real trace instead of the memoryless model traffic. The difference between the two sets of curves is striking. We will not go into the reason behind the shape and properties of the real trace curves, because these have already been tied to the throughput-limitation effect of the protocol. Let us focus however on a related point - if the two chief assumptions about the traffic model which was used are proven to be false (i.e. if the real trace is shown to have ① strong self-similarity in individual stations' traffic streams and ② very unequal distribution of load among stations), then the

statement made in [LIMB 95] no longer applies. This observation is one which has significant implications to an HFC system design engineer implementing a system running the F-CPR protocol (for example), because it shows that simple models will not give true performance indications.

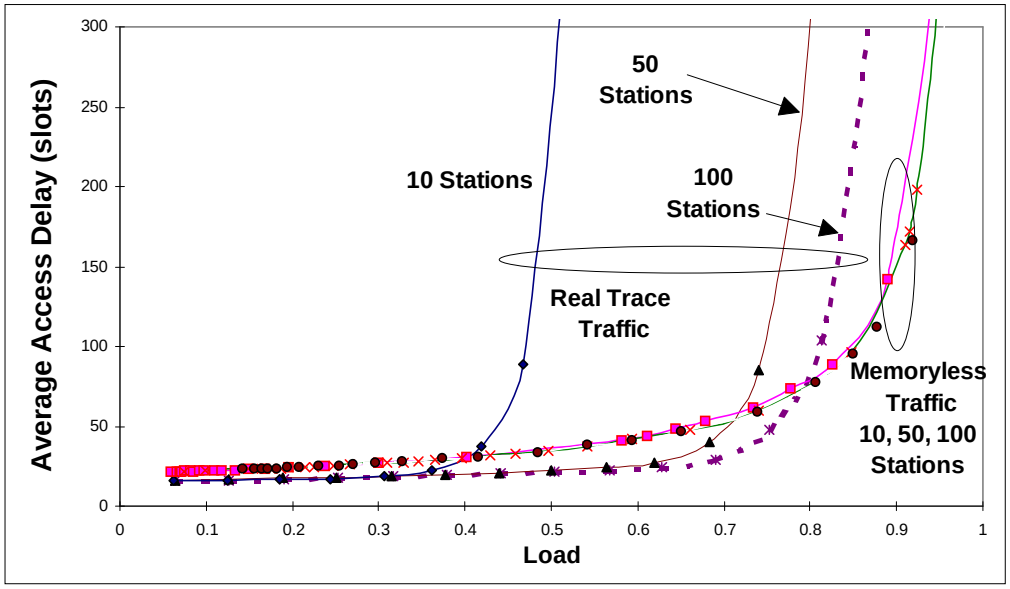


Figure 6.4: Combined effect of Number of Stations and Traffic Type

Average Message Size = 5.961 cells/message

6.2.1.4 Comparison Between F-CPR and CPR

Figure 6.1 to Figure 6.4 has only shown the performance of F-CPR. It may be of interest to the reader how some generic (let's call it "default") implementation of CPR compares to this specific F-CPR implementation. Having tested both the default CPR and F-CPR with a very wide range of different system parameters, from traffic type to fibre and coaxial plant length, we have reached the conclusion that there is *no appreciable difference in any aspect of system performance*, for the two implementations. The two key reasons for this have been found to be (i) a small load range over which significant DMS minislot usage occurs, and (ii) extremely rare cases of simultaneous CMS and DMS arrival at the Head-End, even when DMS minislots are being used.

Expanding on finding (i), we have found that for all combinations of system parameters studied (i.e. model/real trace traffic, system size, and coaxial/fibre cable length), the proportion of total messages transmitted via the use of a DMS slot doesn't even exceed 30% for a load as high as 0.65. This is an interesting finding, suggesting that messages are transmitted almost solely through the contention mechanism for all bar heavy loads, and especially so for uncorrelated (Poissonian) traffic. The obvious result of this is that with so few DMS minislots being used, instances of simultaneous CMS/DMS arrivals at the Head-End are highly improbable and so both the default CPR and F-CPR protocol implementations are more than likely to give near-identical results at these loads.

Finding (ii) is based on the fact that there is an inverse relationship between the proportion of messages transmitted using DMS minislots and those transmitted using CMS minislots. As the load increases, more

and more user station queues are non-empty and the DMS usage rate rises. At the same time, the demand for CMS contention minislots drops, since a station does not need to use these if it already has guaranteed access to the medium (by virtue of its non-empty queue and hence a DMS minislot). As a consequence, even at moderate to very heavy loads, the simultaneous arrival of CMS and DMS minislots remains highly improbable, causing both protocol implementations of CPR to yield identical results.

In summary, and for the reasons given above, we have found that for all possible loads, the likelihood of simultaneous CMS and DMS minislot arrivals at the Head-End for processing is highly unlikely. Since the chief difference between the default CPR and F-CPR implementations lies in the treatment of simultaneously arriving CMS and DMS minislots, it is obvious that the lack of such arrivals will almost eliminate any difference between the two implementations.

6.2.2 Comparison Between F-CPR and Ideal Multiplexer (IM)

An ideal outcome would be for the protocol to perform like an IM, based on maxmin fairness, following the IM and fairness guidelines presented in Chapter 3. In other words, such an Ideal Multiplexer can serve as a *performance benchmark* for the protocol. It is therefore of interest to compare the performance of the protocol with that of such a multiplexer. In Figure 6.5 and Figure 6.6 (overleaf) we compare the utilisation vs. load performance of the F-CPR protocol and that of an IM, for 10 and for 50 active stations respectively.

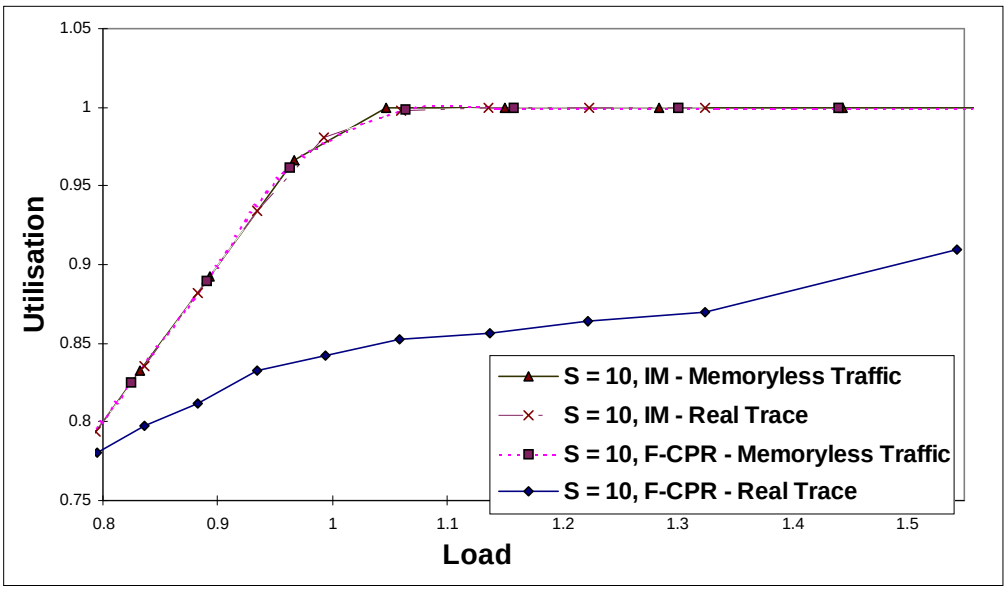


Figure 6.5: F-CPR vs. Ideal Multiplexer - Utilisation Curves for 10 Stations

From Figure 6.6 we can see that for the 50 station case, the protocol performs like an IM, regardless of the type of traffic it is loaded by. The 10 station case is somewhat different - as illustrated in Figure 6.5, unless memoryless traffic constitutes the load, the protocol utilisation is less than that of the equivalent IM by up to 0.15 (at a load of 1.0). This is related to the reduction in efficiency which was noted in Figure 6.2 and is attributable to RTD-related protocol characteristics, which are discussed in Section 6.2.5.

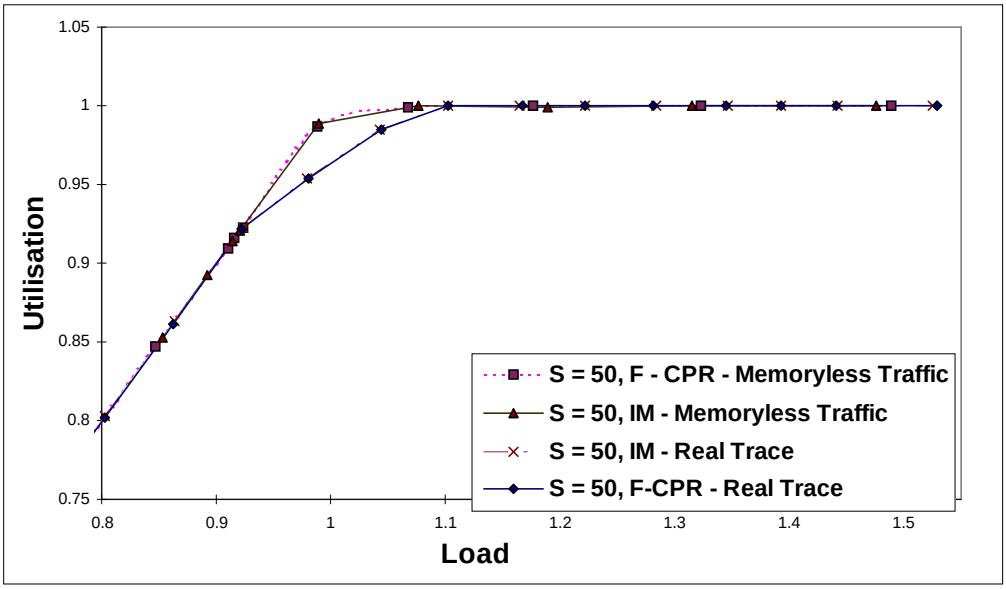


Figure 6.6: F-CPR vs. Ideal Multiplexer - Utilisation Curves for 50 Stations

When comparing the delay performance of F-CPR and that of the IM, in Figure 6.7 and Figure 6.8 (overleaf) for the cases of 10 and 50 active stations respectively, we again observe that the protocol behaves very close to its multiplexer benchmark. In the case of the F-CPR average access delay curves, it should be noted that an unavoidable and constant overhead (8 timeslots in the HFC system configuration we study) manifests itself in the form of fixed propagation and Head-End processing delays, collectively known as the Round trip Delay, RTD. This is something from which the IM, as its name implies, is immune. This first difference between the curves is non-varying with load, and can be effectively ignored, given that all candidate IEEE 802.14 protocols have this exact same "handicap" with regard to the IM.

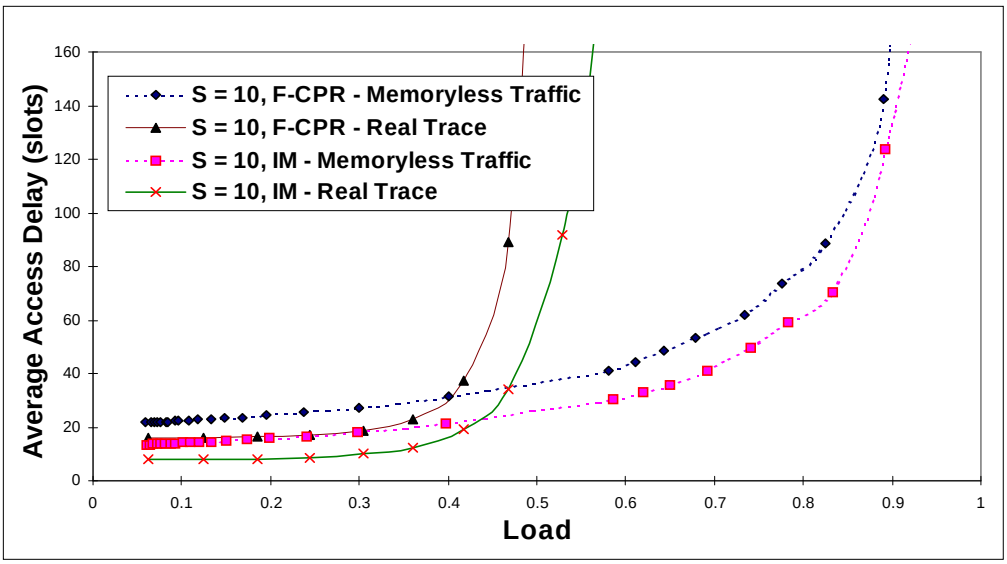


Figure 6.7: F-CPR vs. Ideal Multiplexer - Access Delay Curves for 10 Stations

It is very significant to observe that other than this fixed RTD overhead, there are no noteworthy differences between the IM and F-CPR delay curves, for each of the four F-CPR / IM curve pairs presented in the figures. The chief reasons for this are the lack of any serious contention related delay, and plentiful usage of the collision-free reservation channel. As will be illustrated in the following section,

regardless of the traffic type, even the worst case CMS collision rates do not exceed about 0.15. The only exception is the pair of curves for 10 stations loaded by the real trace, where the average F-CPR access delay seems to actually diverge from that of the IM. This is another manifestation of the F-CPR protocol's throughput limitation effect, mentioned earlier.

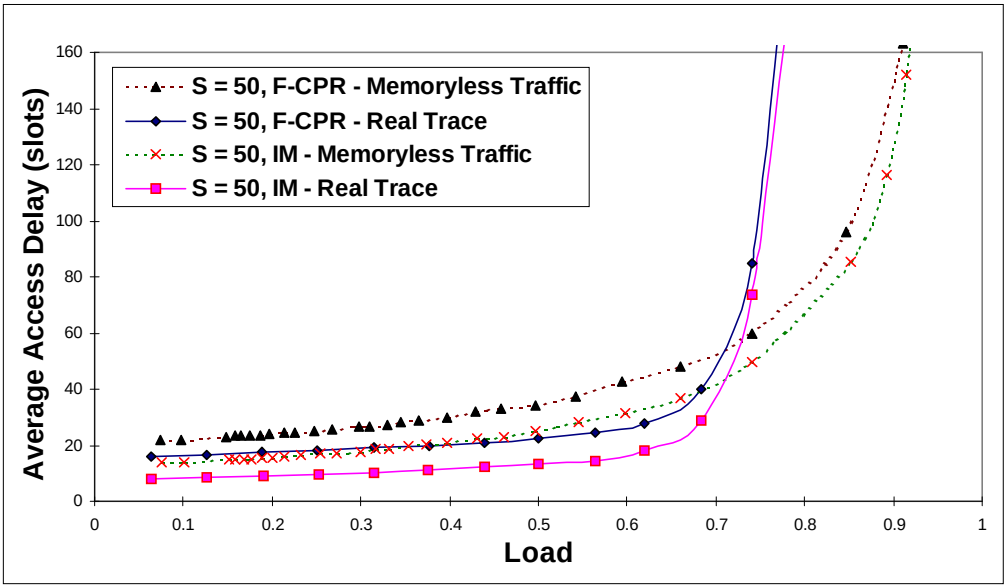


Figure 6.8: F-CPR vs. Ideal Multiplexer - Access Delay Curves for 50 Stations

It is interesting to note that all the observations we have made hold for both the real trace and the memoryless traffic, and that again the memoryless model significantly overestimates performance of both the protocol and that of the IM. Also, consistent with results from Figure 6.5 and Figure 6.6, the *F-CPR converges to IM behaviour (particularly for the real trace traffic) when the number of stations increases from 10 to 50, due to the more evenly spread load and consequently a reduced effect of F-CPR's throughput limitation with the larger system size.*

Two significant factors in the delay performance of the IM (and hence the F-CPR protocol) when loaded by any type of traffic, are: (i) the message size variance, and (ii) the autocorrelation / self-similarity level of the traffic generated by a station. The combination of factors (i) and (ii) is responsible for the system size-independent observation that the memoryless model curves initially exhibit higher delay at low loads, but are then overtaken by the real trace curves at high loads. Namely, factor (i) dominates at the lower loads, when it is the higher variability in message size that causes the memoryless model curves to have higher average access delay. As the system load becomes higher, the highly correlated nature of the arrival process generated by stations reading the real trace starts to dominate the queueing behaviour, until the message size variance is no longer a significant factor. It is important to realise that both factors (i) and (ii) hinge on previously established queueing theory principles ([KLEI 76], [HUAN 95] and references therein) and are a property of any generic single server queue system, rather than of the F-CPR protocol.
Notice that the IM delay performance is unaffected by system size when the input traffic is memoryless, yet it is visibly better in the case of a larger system when the input traffic is a real trace. This comes about as a result of breaking up the original trace's correlated traffic stream into many component streams, when it is read by 50 instead of 10 stations. The overall traffic profile is smoothed by this, since the original

bursts are split up into many smaller bursts occurring at different times. Since the memoryless model generates independent streams without autocorrelation anyway, the number of stations plays no part.

6.2.3 Fairness Testing

Figure 6.9 demonstrates that *maxmin* fairness (defined in Section 3.1.2) is achieved when the traffic load to the F-CPR protocol is a real trace. Furthermore, the same figure shows us that the F-CPR protocol performs near-identically to the IM in this respect. Figure 6.9 depicts a system with 50 active stations, uniformly distributed over a coaxial line distance of six slots (i.e. the distance from the nearest station to the furthest station). The traffic input used is the trace *"kp176half.dat"* replayed at a speed causing overload (load at ~1.5). It is important to ascertain F-CPR fairness under overload conditions, since in practice most data networks operate close to full load, and even exceed their capacity frequently [SALA 96a]. The stations are numbered in ascending order, so that number 1 is closest to the Head-End and number 50 is at the end of the line.

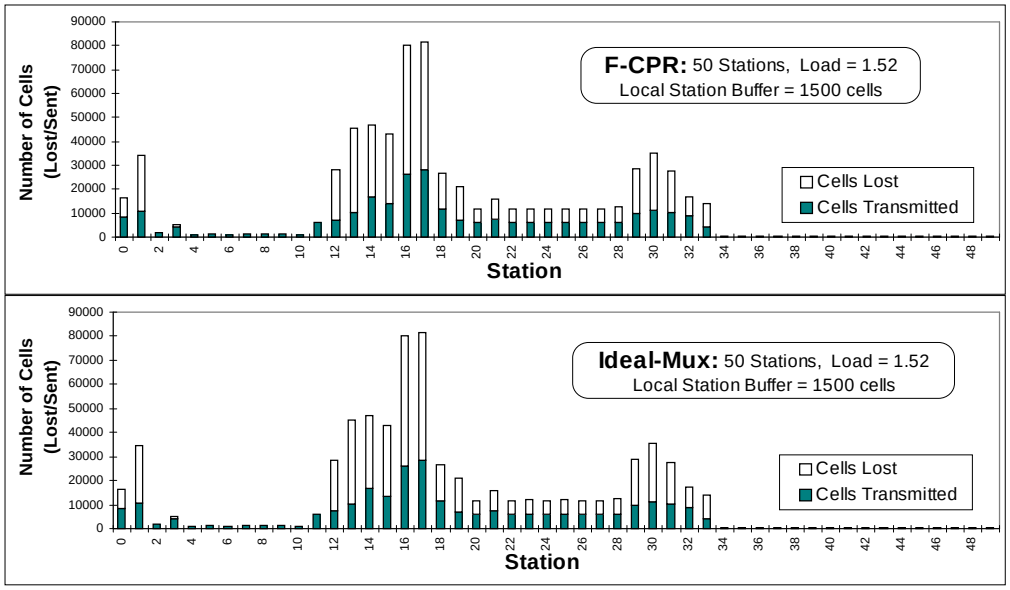


Figure 6.9: F-CPR Maxmin Fairness Test - 50 Stations with local Queue Buffers Limited to 1500 Cells

The criteria we use for determining protocol fairness are (i) the number of cells sent by a station during the entire simulation run and (ii) the number of cells lost during the entire simulation run. The addition of items (i) and (ii) yields the total number of cells which arrived at the station during the simulation run (in Figure 6.9, this would be the combined height of each grey and white column). Note that the shape and values shown in both the upper (F-CPR) and lower (Ideal-Mux) graphs in the figure, are almost exactly the same. Both graphs in Figure 6.9 reflect, in an identical fashion, the classical properties of *maxmin* fairness: light users get the bandwidth they need (with little or no cell loss), while the increasingly heavy users are prone to increasing levels of cell loss.

The other aspect of the F-CPR protocol's fairness testing is related to the relationship between the amount of bandwidth available to any given station, and that station's position along the coaxial distribution line. In Figure 6.10 we once again have 50 active stations, uniformly distributed over a coaxial line distance of six slots.

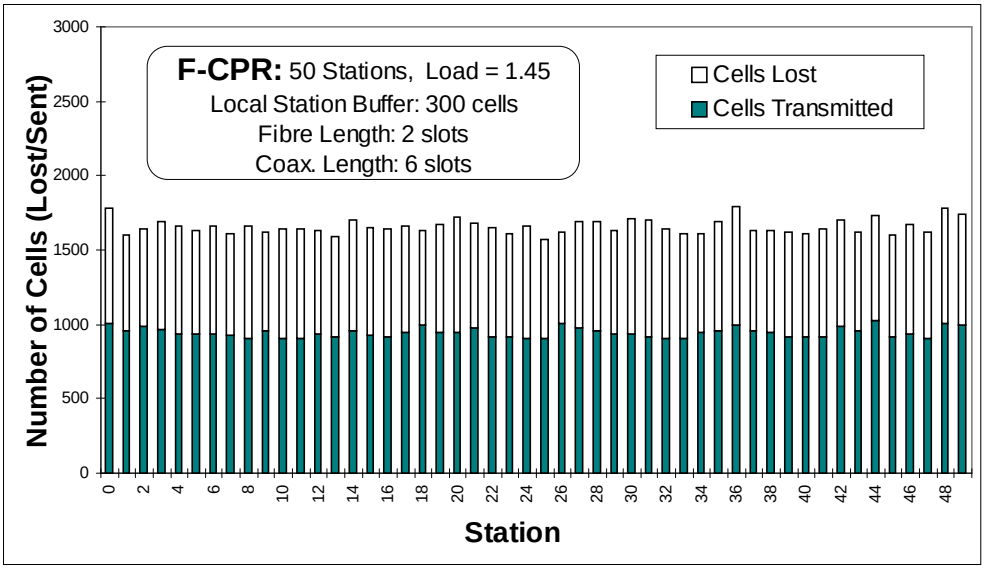


Figure 6.10: F-CPR Positional Advantage Test - 50 Stations, Queues Limited to 300 Cells

However, in order to more clearly focus the positional advantage test, we try to effectively eliminate the differences in the amount of traffic arriving at each station. We have already checked how the protocol treats heavy and low usage stations relative to each other; now we make the incoming traffic more or less equal for each station, and observe whether all stations gain an equitable share of the bandwidth, regardless of their position on the coaxial distribution feed. Note that since we are dealing with a real traffic trace, it is not as easy to obtain identical arrival patterns at each station, as it was in the case of the memoryless traffic used to confirm positional fairness in [SALA 96a]. Instead, we had to choose a portion of the original trace, "*kp176.dat*" with a roughly even activity level, and subdivide this trace among the 50 stations. This new traffic input (the trace "*kp176flat.dat"*) therefore has similar but very slightly different offered traffic for each station, as is evident from Figure 6.10. The trace is again replayed at a speed causing overload (load at ~1.45).

As before, the criteria we use for determining protocol fairness are (i) the number of cells transmitted by a station and (ii) the number of cells lost at the station. Except for the slightly varying offered traffic load, which has been explained to be solely an artefact of the arrival process (i.e. the trace), Figure 6.10 clearly shows the amount of relative bandwidth seized by a given station, to be totally independent of its position on the shared medium. This is consistent with the results of [SALA 96a] which came to the same conclusion for memoryless traffic.

6.2.4 Performance of F-CPR Signalling and Reservation Channels

6.2.4.1 Effect of the Number of Stations

Figure 6.11 shows the CMS collision rate versus load, under memoryless and real trace traffic, for both system sizes. This collision rate is measured as the ratio of collided CMS minislots to the total number of CMS minislots transmitted. The first observation we can make is that as expected, the more stations we have in the system, the higher the number of collisions. Secondly, a significant drop-off in CMS collision rate occurs at some load threshold, due to the filling of stations' queues and the increased usage of (reserved / collision-free) DMS minislots.

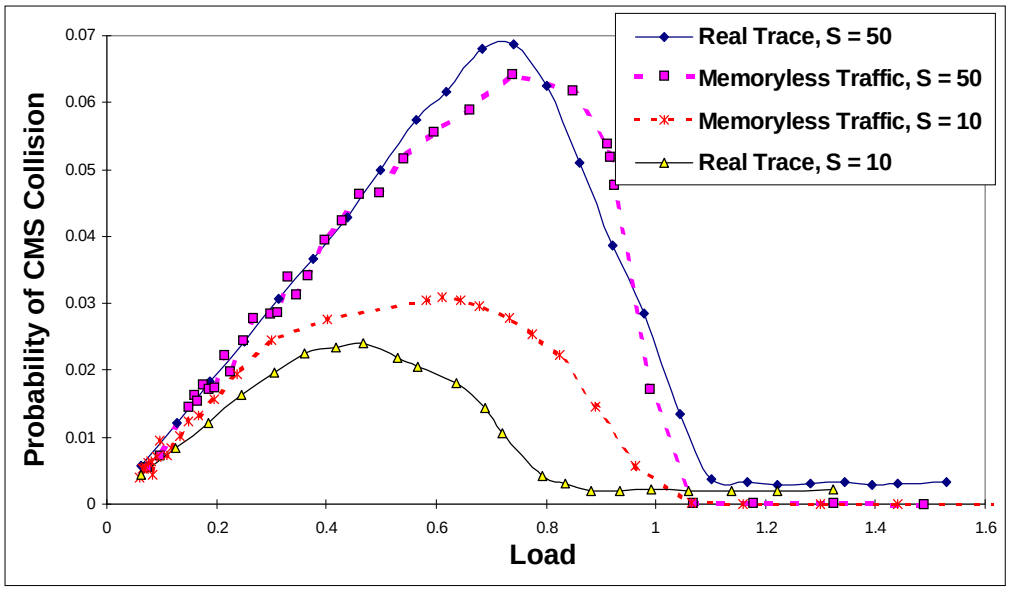


Figure 6.11: CMS Collision Rate versus Load

A point of interest is the observed horizontal displacement of the peak CMS collision rate value, between the curves for 10 stations and 50 stations, in both the case of memoryless and real traffic. That is, the load at which the peak CMS collision rate occurs, is increased substantially as we go from a 10 to a 50 station system. For memoryless traffic the lateral shift is from 0.7 to 0.9 (29%), while for the real trace the shift is much more pronounced, being from a load of 0.5 to a load of 0.75 (50%). As stated, the reason behind the larger systems' curves being higher is the increase in the number of transmitters and hence a higher probability of collisions. This explanation also gives insight into the observed lateral displacement of the peaks for these larger systems. That is, the larger number of transmitters causes a more evenly spread system loading and means that it takes a substantially higher total system load to achieve enough continually (or near-continually) non-empty queues on average - a *critical mass* of such queues. Once this critical mass of such queues has been reached and exceeded (i.e. load exceeds a threshold value), the effect of collision-free DMS minislots begins to outweigh and reduce the otherwise intuitive effect of greater system load (i.e. that greater system load should ordinarily, without the DMS reservation channel, cause more contention for the channel and hence a higher CMS collision rate).

6.2.4.2 Effect of Load Distribution among Station Queues - Plateau Regions

Another point of interest is the plateau regions seen only in the real trace curves in both Figure 6.11 and Figure 6.12, the latter of which shows the DMS usage rate.

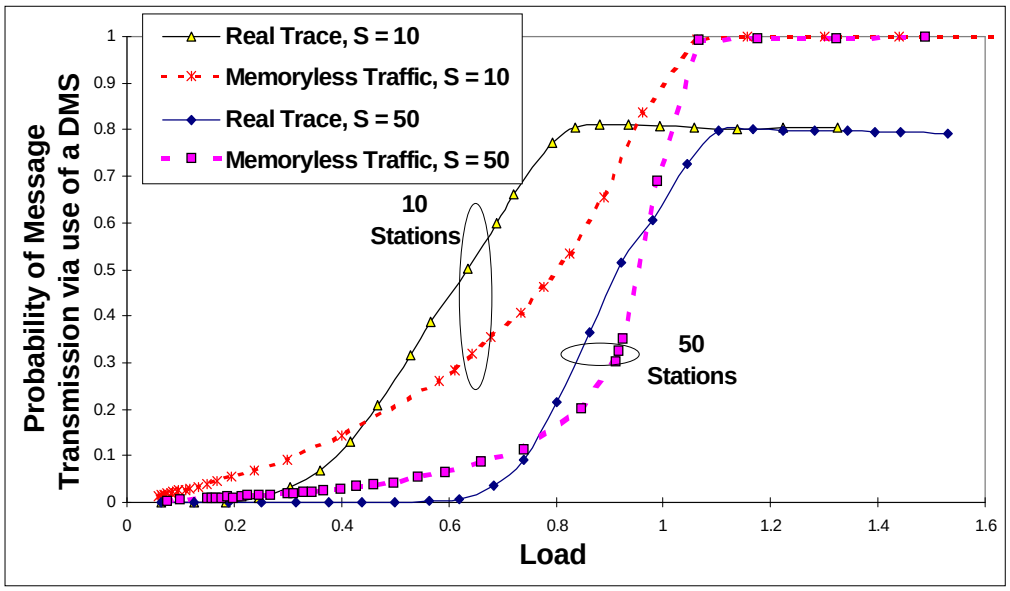


Figure 6.12: DMS Usage Rate versus Load

In the 10 station system, the plateau region for both the CMS collision rate curve and the DMS usage rate curve begins at a load of about 0.8, while in the 50 station system it starts at a load of about 1.1. We discuss these plateau regions together because they are strongly interrelated, since they arise as a result of the same property of the real traffic trace. Starting with the DMS usage rate shown in Figure 6.12, the memoryless model curves will ultimately tend to a DMS usage rate of 1.0. This is interpreted as a state where every message transmission comes about through the use of only the collision-free reservation channel, and there are no collisions possible since the contention channel is unused. This happens due to the very balanced distribution of total system load amongst the memoryless model stations, which in turn causes an identical level of overload at **every** station's queue and allows all stations to (almost) continuously use only the DMS rather than CMS minislots.

On the other hand, if we focus our attention on the real trace DMS usage rate curves in Figure 6.12, it becomes apparent that there is a distinct "plateau region" at a DMS usage rate of about 0.8, which is reached at different load points for the 10 station and 50 stations systems (as quoted above), and cannot be exceeded. This behaviour is a manifestation of the very unbalanced load distribution for the systems using a real trace. Namely, the unbalanced nature of the load distribution of the real trace among stations, causes a small number of users (3 in the 10 station case, 13 in the 50 station case) to overload the system, and leaves many other stations with queues in an underloaded state and hence needing to use the contention channel to access the shared medium every time they register a message arrival.

This residual need to use the contention channel even under heavy system-overload is also clearly evident from the real trace CMS collision rate plateaux observed in Figure 6.11, which commence at the same

load points as the DMS usage rate plateaux. Note that this *residual CMS collision rate* of the 50 station system is slightly higher than that of the 10 station system, being 0.0025 compared to 0.0020, due to the greater number of stations with underloaded queues potentially having to signal for access (37 instead of 7). Finally, corresponding to the observation that the memoryless model traffic tends to be solely carried by use of the reservation channel (i.e. DMS usage rate $\Rightarrow$ 1.0), is the fact that the CMS collision rate of both the 10 and 50 station memoryless model system tends to zero as the load exceeds 1.0.

Note that the exact level of the load distribution imbalance is something which will vary from one trace to the next, and so the heights of both the DMS usage rate and CMS collision rate plateaux will also vary from one trace sample to the next. The ultimate message here is that the memoryless model once again overestimates performance, in that it optimistically assumes that at large system overload, all stations' queues are evenly overloaded and hence no longer need to use the contention-based signalling channel. The memoryless model does not take into account that a very skewed load distribution may cause two or three users to overload the system, leaving many other stations in an underloaded state and hence needing to use the contention channel to periodically access the shared medium.

6.2.4.3 Combined Effects of Message Size Variance and Traffic Correlation Properties

6.2.4.3.1 First Cross-Over Load Point

One of the things immediately noticed when looking at Figure 6.12 is that there is a load point at which the memoryless model and real trace DMS usage rate curves intersect; at 0.45 for 10 stations, and 0.75 for 50 stations. Prior to this cross-over point, for either system size, it is the memoryless model traffic which enjoys a higher usage rate of DMS minislots; and, after the cross-over, the real trace traffic DMS usage rate curves exceed (and increase at a more rapid pace than) their memoryless model counterparts.

This cross-over is directly attributable to the same IM queueing behaviour observations made in Section 6.2.2 for the two system sizes, when loaded by the different traffic types. Initially the memoryless model's higher message size variability dominates the queueing delay at loads lower than the cross-over loads mentioned above. At these loads, it is the memoryless model system which will have a higher system-wide probability of non-empty queues. This is reflected both (i) in Figure 6.12 as a better utilisation of the reservation channel (higher DMS usage rate) than for the real trace system; and, (ii) in Figure 6.7 / Figure 6.8 as a worse delay performance than that of the real trace system. It is interesting to note that at a load of 0.45 for a 10 user system, and at a load of 0.75 for a 50 user system, the roles are reversed - in Figure 6.7 / Figure 6.8, it is exactly at these load points we see the real trace delay performance become worse than that of the model traffic, while at the same time in Figure 6.12, the DMS usage rate of the real trace systems exceed that of the memoryless model systems. After this cross-over, the highly correlated nature of the arrival process generated by stations reading the real trace starts to dominate the queueing behaviour, making the message size variance less and less of a significant factor, and causing the real trace system to have a higher system-wide probability of non-empty queues. In this way, highly correlated realistic traffic, which would intuitively be expected to adversely affect Ideal Multiplexer (IM)

performance in comparison to Poisson traffic, actually somewhat improves the performance of F-CPR by increasing the DMS usage rate!

6.2.4.3.2 Second Cross-Over Load Point

Another observation that may be made from Figure 6.12, is that each pair of 10 station and 50 station curves has yet another cross-over point for the DMS usage curves. The reason for this subsequent cross-over may be inferred from the explanation about the origins of the "plateau regions" in Section 6.2.4.2, with the memoryless model curves showing an even system-wide overload of station queues and hence tending to a DMS usage rate of 1.0. Meanwhile, the real trace curves cross their memoryless model counterparts at both system sizes, and hit an asymptotic limit at a DMS usage rate of 0.8, this time not because of IM-related queueing behaviour, but due to the extremely unbalanced nature of the system load amongst member stations preventing any greater level of DMS minislot usage.

6.2.4.3.3 Greater Impact of Traffic Self-Similarity for Smaller Systems

One of the things noted from the IM delay curves in Figure 6.7 and Figure 6.8 was that the real trace 10 station system performed much worse than the 50 station system (going to unusable levels at a load as small as 0.45 instead of 0.75), while the memoryless model system had delay performance curves unaffected by the number of transmitting stations. The reason for this is that the limiting factor at higher loads is no longer message size variance; rather, it becomes the level of long-range traffic autocorrelation, or self-similarity. Being memoryless, the Bernoulli-Geometric model which we use has already been shown not to be self-similar at all, so the number of stations generating the traffic is an irrelevant factor, and we see this via the identical delay performance of both the 10 and 50 stations memoryless model curves.

On the other hand, remembering the way the real trace file is serially divided up and the resulting parts are assigned to the stations (Sections 2.6.2), it is obvious that the system with only 10 stations will result in five times as many real trace file pieces, which are five times as long as the ones used in the 50 station system. This increased length means that the **stations in the smaller system retain much more of the correlation properties of the original real trace.** The detrimental way that this impacts the queueing behaviour when these real trace systems load the IM is as stated above - delay performance is seriously worsened when the smaller system loads the IM. From the graphs in Figure 6.7 and Figure 6.8, we see that this results in a situation where the 50 station real trace system has a usable queueing delay load-limit of 0.75, which compares quite well to the load-limit value of 0.95 for the memoryless model system of the same size. This is a much closer match than is the case with the smaller system (real trace has a limit of 0.45 and the memoryless model has a limit of 0.95). The fact that the larger pair of systems is much more similar in queueing delay behaviour than the smaller pair, is clearly reflected in both Figure 6.11 and Figure 6.12. These figures show that both the CMS collision and DMS usage rates of the 50 station real trace and memoryless model systems are much closer than those of the 10 station systems. The impact of the worse queueing behaviour, and hence "fuller queues more often" for the 10 station real trace system is exhibited (quite counterintuitively!) in the form of a significantly smaller CMS collision rate curve and larger DMS usage rate curve, than those of its memoryless model counterpart.

As mentioned previously, what is usually an adverse traffic characteristic in terms of queueing performance (i.e. high stream correlation), here helps to improve protocol performance.

Finally, a note regarding the observed results needs to be made, as they relate to the way in which the real trace read process was implemented. It has already been explained that the larger system of 50 stations (feeding from the same correlated real trace as the smaller system of 10 stations) divided the real trace up into smaller components and thus reduced both the level of correlation and the level of load imbalance. This helped to alleviate the RTD-related throughput limitation to a certain extent (see 6.2.5.3 in the next section) and so improved F-CPR performance, and brought it much closer in line with that of an IM. However, a word of warning is in order here - this is an effect which has occurred as a result of the way we have implemented the real trace read process. *It is quite possible to have certain situations in real HFC systems where the fact that a system is larger will not necessarily mean a more even spread of load, nor a less correlated individual traffic stream from each station - offering no such immunity from potential throughput limitation problems.*

6.2.5 Insight into Loss of Bandwidth Utilisation Due to F-CPR Protocol Characteristics

In this section we explain the reason for the reduced efficiency of the F-CPR protocol as compared to that of the IM, and highlight certain conditions which may worsen this reduction. In Figure 6.5, it was seen that the channel utilisation for the real traffic trace and 10 active stations using F-CPR, at one stage fell to almost 0.15 below the IM utilisation, as the load began to exceed 1.0. The worse utilisation and delay performance of the F-CPR protocol compared to the IM is particularly pronounced for *strongly correlated real traffic traces*, with *only a few stations* accessing the medium but *supplying high overall system load*. This may be attributed to a couple of contrasting characteristics of the F-CPR protocol and the IM, which we discuss below and illustrate with an example.

In our simulation model, the IM consists of a number of queues (representing stations) all serviced by an ideal round-robin server. The server has the capability to take a cell from any given queue, immediately upon its arrival - within the same slot time. In other words, during the very next timeslot after a just-completed burst from source *x*, it will begin to serve the first cell of source *y*'s new burst. As the name suggests, the round-robin server will serve the queues of the stations one after the other, in a round-robin fashion starting at the station with the middle ID number and continuing onto the others in descending order (and doing a *wrap-around* from the lowest ID to the highest ID, each time it hits the end). To ensure server fairness, each station is served exactly once during each all-station service-cycle, meaning that once a station has had a message processed out of its queue, it will not be afforded service until all other stations (with non-empty queues) have been given the same opportunity. The F-CPR protocol has the overheads of signalling (contention- and reservation-based) and RTD, which prevent it from being a true ideal multiplexer. However, this extra delay, being of a non-varying type, was considered and accounted for in the delay comparisons between the protocol and its IM benchmark. But as was shown in Figure 6.7, in the case of heavy real trace system load with 10 stations, the average F-CPR access delay curve seemed

to diverge from that of the IM. Hence, not only did we fail to observe a constant difference between the two curves, but we also did not see any convergence at heavy overload, (as was noted for 50 stations and all memoryless traffic curves). Thus, not even the protocol's added overheads of signalling, propagation, and processing/response delays could account for such an observation. The answer for the protocol's worse delay and utilisation performance under the conditions just described lies in the ***go/stop*** nature of individual station transmissions. Although a heavily loaded station doesn't have to contend again if it has more than one message in its queue, it still has to *stop* and *wait* for its next reserved allocation, upon the completed transmission of *every* enqueued message (hence we call this the *go/stop phenomenon*). The station sends the last cell along with a request, and then *stops* transmitting. The quickest-response scenario is that two propagation delays and a processing time later (i.e. a fixed delay equal to *RTD*), a grant comes along. In the meantime, all *RTD* data slots travel upstream unused unless:

- another station(s) has *previously* requested (either by contention or reservation) an allocation of slots longer than this *RTD* slot gap, and been scheduled to completely "fill-in" the gap, by the Head-End.

- another station(s) has *previously* requested (either by contention or reservation) an allocation of slots less than this *RTD* slot gap, in which case the gap is partially filled.

This protocol characteristic is best illustrated by an example.

6.2.5.1 F-CPR Go/Stop Inefficiency: An Example

Figure 6.13 (overleaf) shows the size in cells of arriving messages, for the three active stations (out of a system population of 10) in the first 400 timeslots of the real traffic trace *"kp176.dat"*, under heavy overload conditions (Load = 2.0). Note that the arrival statistics for the other seven stations are not shown, because between all of them during the period of observation, they did not even generate a total of ten cells of traffic, and thus had near-minimal impact on the protocol operation (especially with regards to the example of inserted silences, which is our main focus here).

The arrival statistics shown in Figure 6.13 are such that there are two stations significantly loading the system (station #1, black bars, and station #4, white bars giving approximately 80% of total system loading), another medium-load station (station #3, grey bars, providing about 19% of total system loading), and the other seven stations providing a minimal proportion of the aggregate load (1% or less). While the exact proportions of the load provided by the stations may not be wholly realistic, this type of traffic trace for the protocol is a valid scenario, because the concept of small high-usage groups of users and large but relatively quiet users, is nothing new in user population traffic modelling.

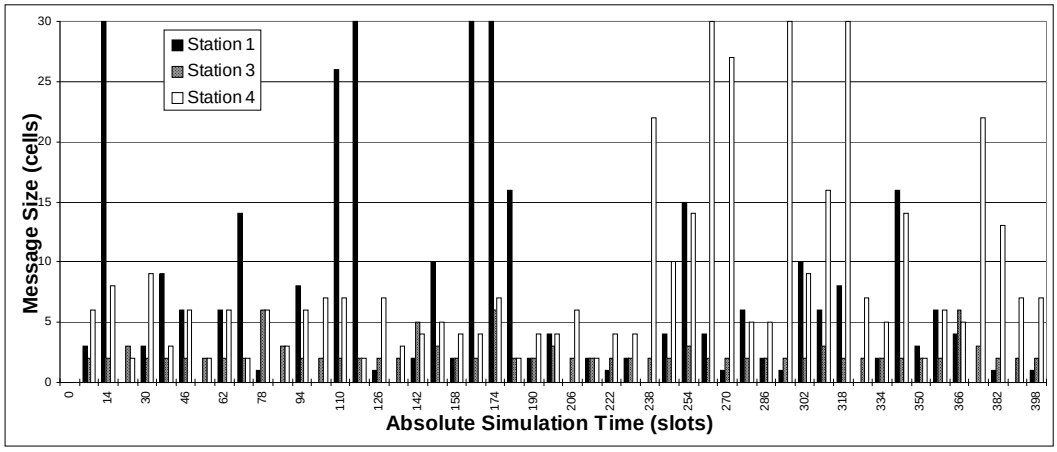


Figure 6.13: Message Arrivals for System Load = 2.0, Total Stations = 10
Active Stations Shown (ID# 1, 3 and 4)

In order to illustrate the inefficiency associated with the protocol's go/stop phenomenon as benchmarked against the IM's round-robin server, we present the first two recorded message arrival instants in Table 6.1, and in Table 6.2 (overleaf) we illustrate the line state when the same traffic trace is applied to both the F-CPR protocol and the IM. Note that, in order to simplify the example the F-CPR arrival process algorithm was modified so that, during the period of observation (400 slots), all arrival events occurred simultaneously at certain time slots (e.g. 6, 14, 22, 30 ...), and were not permitted to occur at any other times.

Absolute Simulation Time (slots)	***Message Arrivals (station ID and quantity in cells)***
6	*Station 1: 3 cells* *Station 3: 2 cells* *Station 4: 6 cells*
14	*Station 1: 30 cells* *Station 3: 2 cells* *Station 4: 8 cells*

Table 6.1: Recorded Message Arrivals for Active Stations 1, 3 and 4 (in cells)

The numbers in Table 6.2 represent the station IDs, so for example, station #1's message of three cells will appear as "1 1 1" in the diagram. Unused data slots are marked with the letter "E".

F-CPR Protocol - Time and Line IDs	*Ideal Multiplexer - Time and Line IDs*
t = 0	**t = 0**
E E E E E E E E E E E E E E E E E E E E E E	***E E E E E E*** 4 4 4 4 4 4 3 3 1 1 1 9 4 4 4 4 4 4 4 4 3
E 9 ***E E E E E E E E*** 3 3 4 4 4 4 4 4 4 1 1 1 3 3 ***E E***	3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
E 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 4 4 3 3 3 1 1 1 4 4 4 4 4 4 4 4 4 4 3 3 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 4 4 10 10 ***E*** 1 1 1 6 6	1 1 1 10 10 6 6 4 4 4 3 3 1 1 1 1 1
t = 100	**t = 100**
3 3 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 3 3 4 4 4 ***E E***	1 4 4 4 4 4 4 4 3 3 1 1 1 1 1 1 4 4 3 3 1 1 1 1 1 1 1 1 1
E 1 1 1 1 1 1 1 3 3 4 4 4 4 4 4 4 1 1 1 1 1 1 1 3 3 4 4 ***E***	1 1 1 1 1 4 4 4 4 4 4 4 3 3 1 4 4 3 3 1 1 1 1 1 1 1 1 7 7
E E E 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 7 7 7 4 4 4 4	7 4 4 4 4 4 4 4 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 1 3 3 7 7 7 5 ***E*** 4 4 1 1 1 1 1 1 1 1 3	1 1 1 1 1 1 1 1 1 1 1 7 7 7 5 4
t = 200	**t = 200**
3 3 3 3 3 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	4 4 3 3 3 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 2 3 3 3 9 9 9 4 4 4 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 9 9 9 4 4 4 4 4 4 4 3 3 2 2 2 2 1 9 4 4 4
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 4 4	4 4 4 4 3 3 1 1 4 4 4 4 4 4 4 4 3 3 1 1 1 1 1 1 1 1 1 1 4
4 4 4 4 9 2 2 2 2 1 3 3 4 4 4 4 4 4 4	4 3 3 1 1 4 4 4 4 4 4 4 4 3 3 1 1
t = 300	**t = 300**
1 1 3 3 ***E E E E*** 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 3	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
3 6 6 6 4 4 ***E*** 1 1 3 3 2 2 2 4 4 4 4 4 4 4 4 1 1 1 1 1 1	6 6 6 4 4 4 3 3 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 8 8 8	1 1 1 1 1 1 1 1 1 1 1 1 1 1 10 8 8 8 4 4 4 4 3 3 3 3 3 1 1
3 3 4 4 4 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4

Table 6.2: Channel State for Ideal Multiplexer and F-CPR Protocol under same Traffic

Two important performance differences are immediately discernible from Table 6.1 and Table 6.2:

a) At Simulation Start-up ...

Unlike the multiplexer, which processes the first upstream data slot at t = 6, F-CPR does not do so until t = 23 (when station #9 transmits its one cell message). This clearly illustrates the inefficiency due to signalling and contention, whereby stations #1, #3 and #4 suffer a simultaneous collision (at t = 6) and then proceed with the p-persistence algorithm (more collisions, but only two of the stations are involved) until one of them finally succeeds (station #3 at t = 32). The multiplexer merely processes the stations in a round robin fashion (stations #4 then #3 then #1, as illustrated) with no collisions, nor any contention resolution needed - since signalling is not needed, giving the IM its “ideality” in that it has full knowledge of the state of each of the stations’ queues at all times.

b) During the simulation ...

The IM, (apart from its initial silence period of six slots, during which no station had any traffic) never allows the channel to have empty upstream data slots. This shows the absence of any signalling, propagation and processing delays, and the channel is "packed" with cells as tightly as is possible.

In the case of the F-CPR however, the go/stop phenomenon immediately begins to rear its head with the appearance of a three slot silence period at t = 45. The reason for this silence lies in the traffic's arrival pattern and the protocol's need to halt a station, even though its queue may be non-empty, in order to

reserve bandwidth for the next transmission. So, given that the F-CPR Head End algorithm scheduled the *t*=6 arrivals (see Table 6.2) of stations #3, #4 and #1 at *t*=[32-33], *t*=[34,39] and *t*=[40-42], it follows that since no other stations recorded arrivals in the meantime, these three active stations were to continue emptying out their queues.

Station #3's last cell was transmitted at t = 33, so taking into account the $RTD = 8$ slots associated with this go/stop phenomenon, plus the one extra slot a station waits to transmit once it's received the grant, the earliest possible time for #3 to continue emptying its queue would have been t = 41+1 = 42. The channel was still busy at this time with #1's transmission, so #3 continued with its next message at t = [43-44]. This is where the silence troubles begin - station #1 finished its last cell at t = 42, so its earliest re-commencement time would have been t = 51. Station #4 on the other hand finished at t = 39, hence its earliest re-commencement time had to be t = 48 (which is what happened, as can be seen from Table 6.2). The time period t = [45-47] thus remained wasted.

Therefore, in this simple example we have seen how a three-slot silence period was inserted because the large system load was being generated by too few stations (only three in this case). Had there been more active stations, the probability would have been higher that a message (or messages) from other stations would have arrived and been scheduled to fill in the *RTD* gap. This type of gap can never happen in the IM because it will begin serving the next non-empty queue, as soon as its current message is over.

It should be said that this type of protocol inefficiency is exacerbated to the extreme when a single station is responsible for sustained, heavy loads over significant periods of time (i.e. no other stations to ever fill in the gaps). Clearly this means instances where we either have very few stations, or extremely highly correlated traffic, or both, as was the case we were investigating here (only 10 stations under highly correlated traffic overload).

6.2.5.2 Effect of Protocol Inefficiency On Average Lengths of Burst and Silence Periods

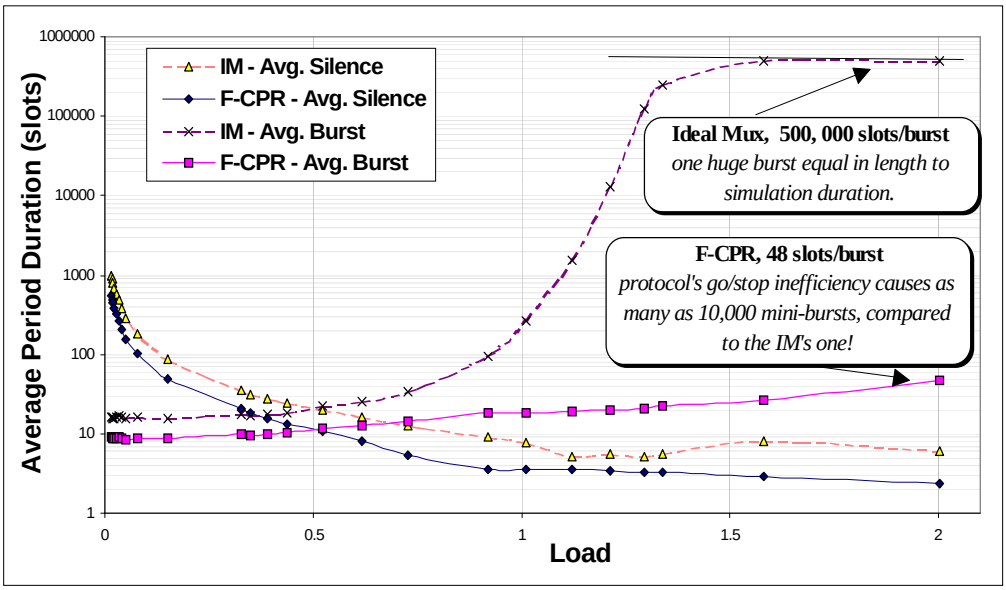


Figure 6.14: Comparison of Average Burst/Silence Period Durations vs. Load - 10 Stations using the real trace *"kp176.dat"*

Figure 6.14 (previous page) and Figure 6.15 were produced by measuring the average *activity burst* and *silence period* durations on the upstream channel, and the number of occurrences of each period. The aim was to illustrate the effect of F-CPR's throughput limiting go/stop behaviour on the usage pattern of the upstream channel, in a configuration which was extreme in that it had a very unbalanced load distribution, a very small number of transmitters, as well as highly correlated traffic streams produced by the individual stations.

The term silence period is self-explanatory, and refers to a block of unused upstream channel slots. An activity burst is defined as an uninterrupted block of continuous upstream slots, in which one or more messages has been scheduled "back-to-back" by the Head-End. It should be pointed out that the number of occurrences of silence and burst periods are necessarily equal so Figure 6.15 may also be interpreted as the number of burst-silence pairs occurring through the simulation. The fixed RTD of the protocol means that an F-CPR burst can never have two contiguous messages from the same station. This is not the case with the IM, which has full knowledge of all stations' queue states, and can rapidly, and without any overhead delay, empty out a given station's queue, if it is the only non-empty one in the system. This behaviour is clearly reflected in Figure 6.14 and Figure 6.15.

In particular, note the way in which the F-CPR and Ideal Multiplexer average burst durations increase with load in Figure 6.14. The IM climbs to an average burst size equal to the entire simulation length at excessive loads (over 1.5), and as Figure 6.15 shows, only one pair of silence-burst periods is recorded. In other words, the IM is optimally efficient in that it generates one extremely long burst with 499,994 time slots (virtually 100% of simulation length) leaving a very short silence period of 6 slots. On the other hand, in Figure 6.14 it may be noted that even for very large loads the F-CPR protocol's average burst size increases only moderately compared to its low-load levels, to ultimately reach about 48 slots at a load of 2.0. Furthermore, the upper right-most tip of Figure 6.15 shows that this burst size of 48 slots/burst is averaged over not one, but 10,000 burst periods.

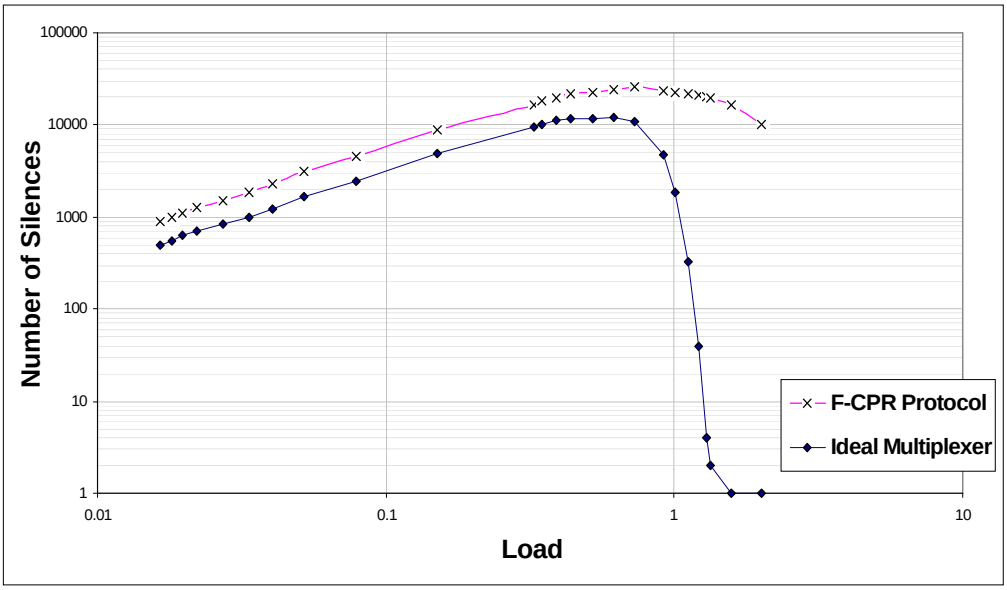


Figure 6.15: Number of Silences vs. Load - 10 Stations using the real trace *"kp176.dat"*

It is clear from the curves in these two figures that the F-CPR protocol is unable to fully utilise the available slots on the channel, due to the go/stop phenomenon described earlier. As illustrated with the example of Section 6.2.5.1, and as shown with figures in this section, the result is that the protocol introduces a larger number of silences onto the transmission medium than its IM counterpart.

6.2.5.3 Properties of Traffic and the HFC System which worsen the Go/Stop Phenomenon

With the chosen reference configuration of the HFC system under study here, the go/stop effect only becomes a particularly noticeable utilisation inefficiency problem at large loads (>0.8) where the effect of a very small number of sources with strongly correlated and unbalanced loads, is highly pronounced (because the message interarrival times become much smaller than *RTD*, within the high-activity stations' queues). The problems associated with a go/stop effect stem largely from the size of the *RTD* in cell slots, as compared to the average message sizes themselves. The larger the *RTD*, the greater the potential utilisation inefficiency of the F-CPR protocol. Three parameters of HFC systems which have a direct impact on the *RTD* length are:

- physical separation of the Head-End and the most distant end-user station.
- transmission speed of the upstream channel.
- Head-End processing time (read, write and scheduling latencies).

An **extreme** RTD case would be a system with (i) a very large distance between Head-End and furthest station, (ii) a very fast upstream transmission speed and (iii) a very long Head-End processing time. In addition to the physical HFC system properties mentioned, it is correlation of the individual stations' traffic streams, the average message size in cells, and the evenness of the system load spread, which have the potential to cause further inefficiency problems through the go/stop phenomenon.

With all of these factors taken into account, one can envisage a worst-case system with an enormous RTD and a single station transmitter sending most of the load, in a very large bursty, correlated fashion. Depending on the actual values of all of the factors described above, the single station would be able to utilise a wide variety of upstream bandwidth values (ranging anywhere from 10% to 90% of total upstream bandwidth). However, with **typical** network sizes and speeds (Head-End to furthest user distance ~ 40km, upstream channels less than 10Mbit/s), and expected average message sizes not being less than at least 3 ATM cells, this figure is unlikely to fall below 70-80% of available upstream bandwidth. Remembering that a single overloaded and highly correlated transmitter is an unlikely and worst-case occurrence in itself, this figure of 70-80% will not pose a practical problem to network efficiency in our opinion. This is especially true, since in such worst-case scenarios there are no other users (or perhaps one or two other ones only) of the upstream channel's resources, and the argument that bandwidth is being wasted is very hollow - given that there is no one else to use it at that time.

6.2.5.4 Multiple Unacknowledged Messages: Overcoming the Go/Stop Phenomenon

The version of CPR presented in [SALA 96a] and its extension into F-CPR in this research work has assumed only one outstanding (unacknowledged) message per station. In order to eradicate the phenomenon of the go/stop nature of the protocol, one would have to allow multiple unacknowledged messages per station. However, this would introduce a trade-off analogous to one described in [SALA 96a], where the need for multiple grant and contention mini-slots was scrutinised, by weighing up the benefits against implementation costs. In the case of trying to eliminate the go/stop effect of F-CPR, the **benefits** are:

- A very small incremental benefit in delay and utilisation performance noticeable only under extreme properties both of the physical HFC system and the traffic being generated, as described in 6.2.5.3.

On the other hand, the **costs** are:

- Increase in station algorithm complexity.
- Significant increase in Head-End scheduler complexity and necessitate a complex bandwidth manager, in order to maintain fairness. Currently the Head-End scheduler simply schedules what it sees and takes no other actions.

Weighing up the costs and benefits, the introduction of multiple unacknowledged messages will arguably not have a significant positive impact.

6.3 Multiple Priority Systems

6.3.1 Effect of Priority Assignment Mechanism on All Priority Scheduling Schemes

The graphs of Figure 6.16 through Figure 6.18, shown over the next three pages, highlight the average access delay for traffic of each priority level for the 10 station F-CPR: JET / SA and IM systems employing the RMP and PG (unshuffled and shuffled) priority assignment mechanisms. After these figures, we present Figure 6.19, which shows the delay for traffic of high priority for the 50 stations systems employing these same priority assignment mechanisms. Surveying the four figures leads to three important observations: (i) regardless of priority assignment mechanism or system size, the JET scheme leads to clearly superior delay performance of high priority messages, at the expense of the lower two priorities; (ii) for the 10 station IM systems, the different priority assignment mechanisms lead to variations in relative access delay performance as we go from one priority to another. That is, the relative positions of the RMP and PG curves change quite significantly for each priority; for example, the unshuffled PG and RMP systems have equal delay curves for low priority, while the RMP has a lower

delay curve for high priority traffic (see Figure 6.16 and Figure 6.18); (iii) the protocol affects the delay performance of the PG scheme loaded by an unshuffled trace more severely than that of the other two schemes, regardless of whether JET or SA is implemented; and, in these affected PG-unshuffled trace systems the medium priority traffic is more adversely impacted than the low priority traffic.

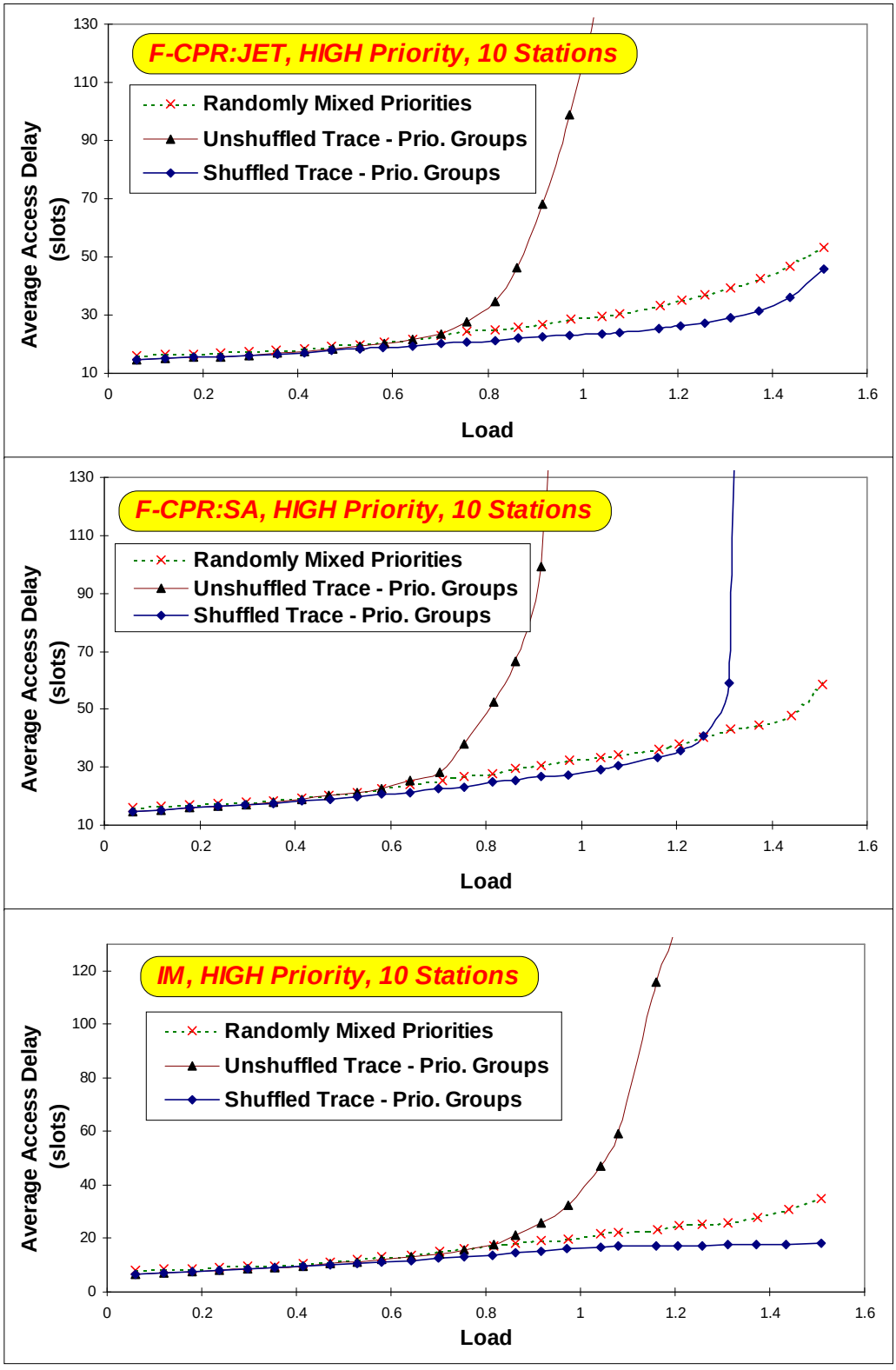


Figure 6.16: Priority Assignment Mechanism Effect - 10 Stations, High Priority

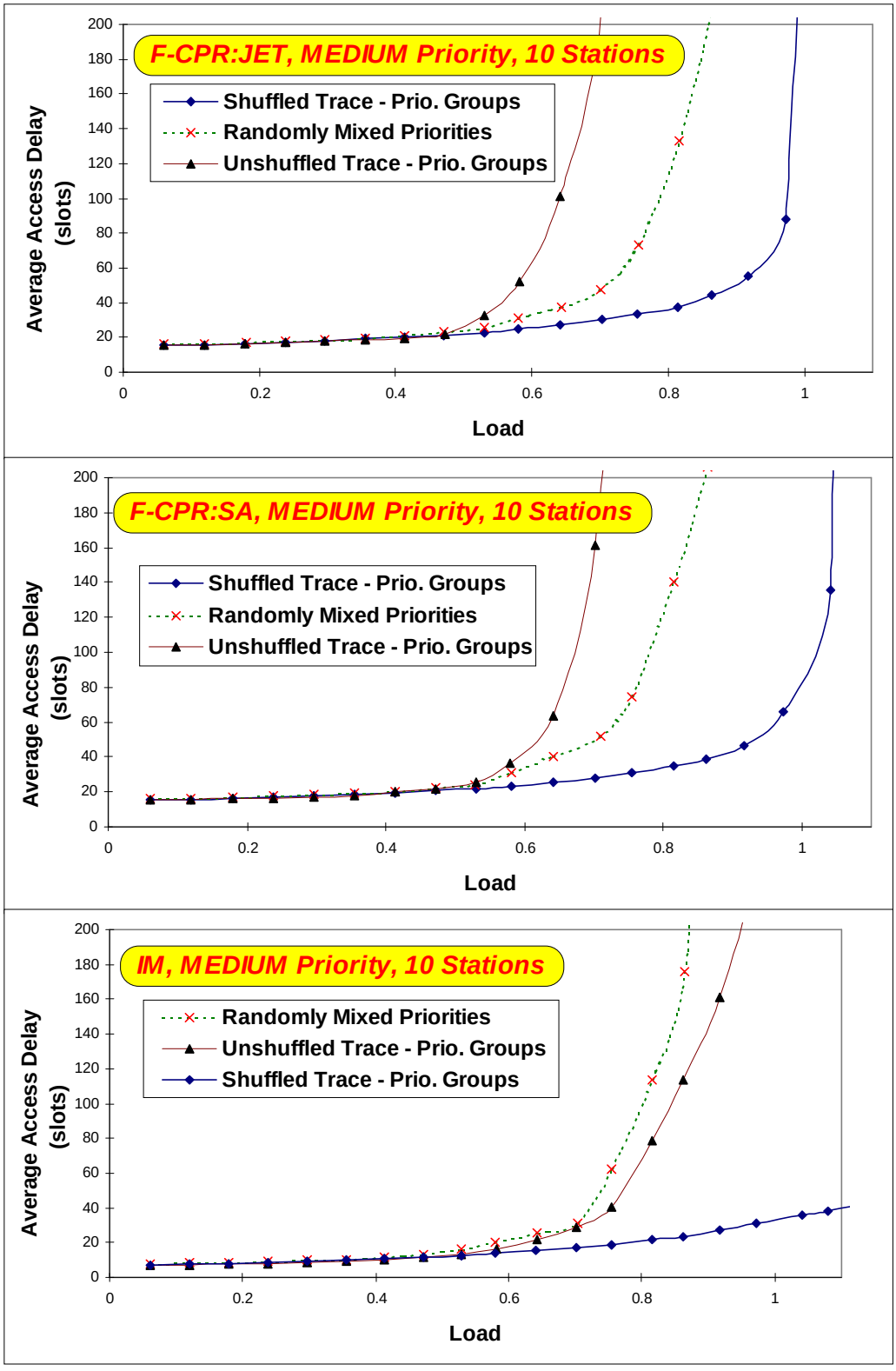


Figure 6.17: Priority Assignment Mechanism Effect - 10 Stations, Medium Priority

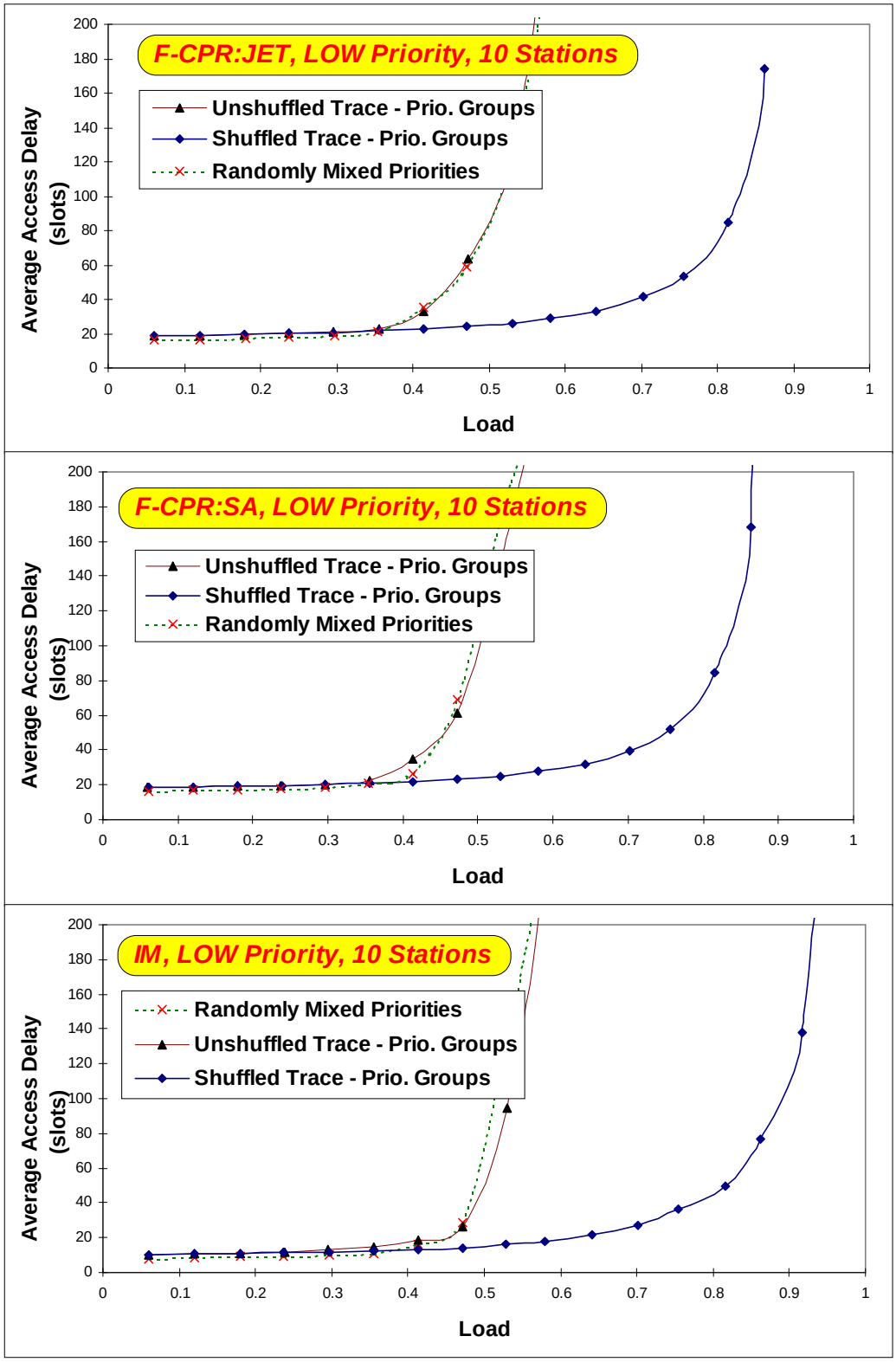


Figure 6.18: Priority Assignment Mechanism Effect - 10 Stations, Low Priority

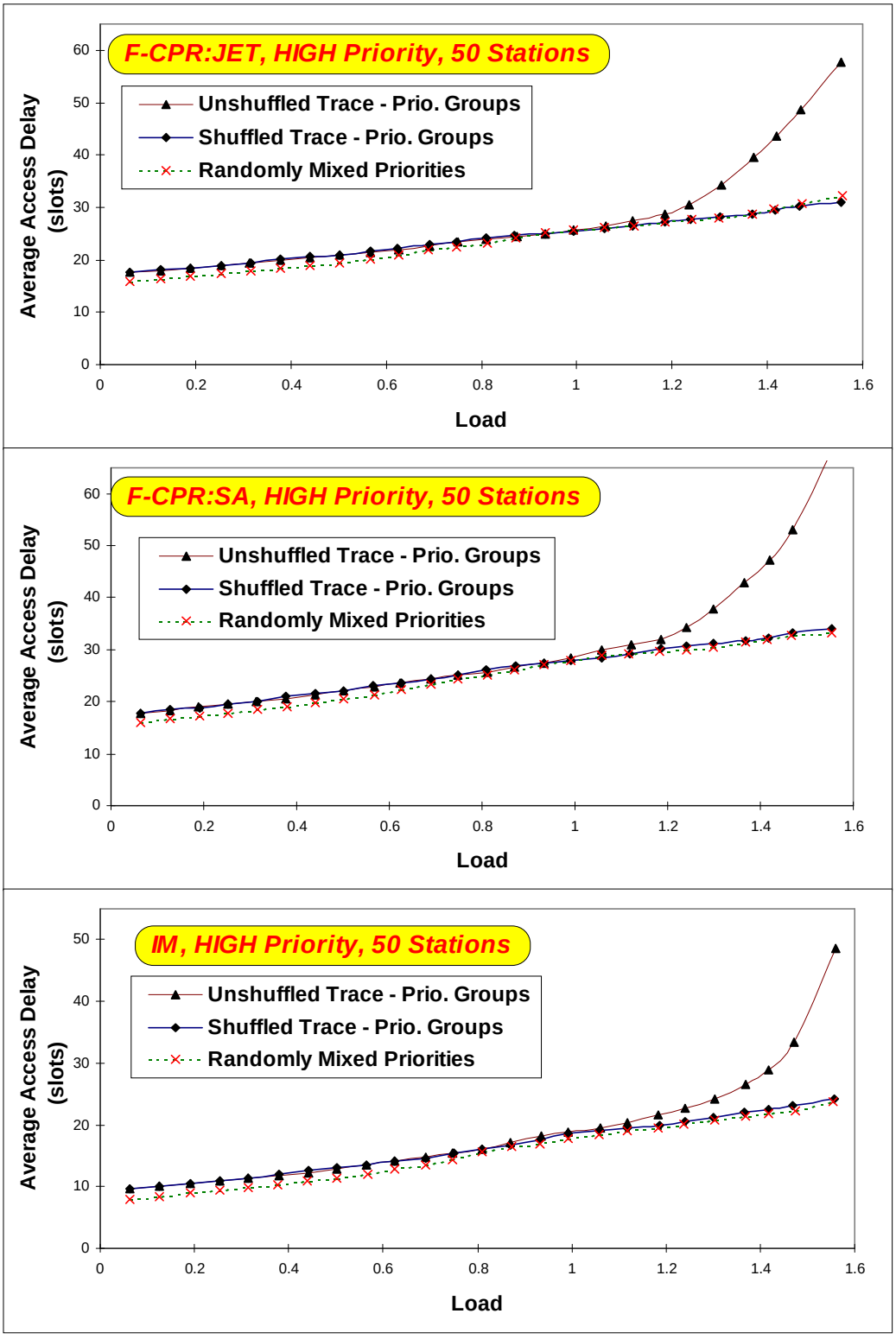


Figure 6.19: Priority Assignment Mechanism Effect - 50 Stations, High Priority

Let us now look more closely at what stands behind each of these three observations, in turn.

6.3.1.1 JET Scheme - Superior Delay Performance compared to the SA Scheme

Firstly, Figure 6.16 illustrates that the ***high*** priority traffic delay performance for the JET scheme is superior to that of the SA scheme, whether the systems are PG or RMP based. The SA scheme's worse

performance in comparison to the JET is quite noticeable for the RMP system (20% higher delay on average), the unshuffled PG system (delay increases to infinity at much smaller load, 0.85 instead of 1.15), and finally, for the shuffled PG system where the JET scheme yields tolerable delays even at a load of 1.5 while SA goes to infinity at a load of only 1.3. Figure 6.19 shows a much closer contest for large system size (50 stations), with the difference in the JET and SA schemes' average access delay almost imperceptible. Related to this superior delay performance, Figure 6.24 illustrates that the high priority traffic is allowed to utilise more of the available channel bandwidth, when the JET scheme is used instead of the SA scheme. The reason for this behaviour is that the JET scheme is based on a pre-emptive delaying of lower class requests. We do not go into the detail of this pre-emption property, since it is the topic of discussion in Section 6.3.2.

6.3.1.2 PG Systems : Dissimilar Traffic Arrival Profiles per Priority

The second observation relates only to systems with 10 stations, and comes about as a result of the different characteristics of the input message streams for each priority of the supplied system load, as shown by the pie charts in Figure 6.20.

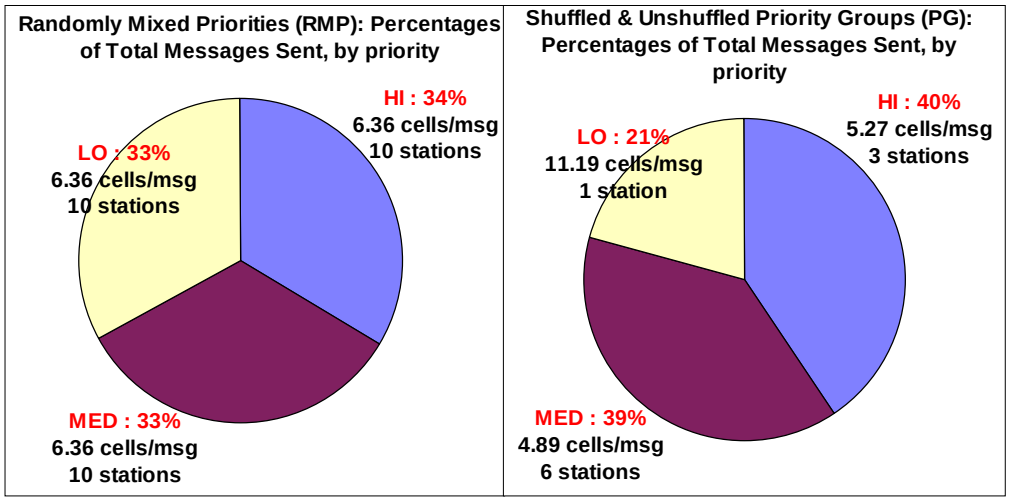


Figure 6.20: Per-Priority Traffic Profiles for 10 station RMP and PG Systems

For RMP systems it was easy to generate equal loads (measured as total cells generated) by all three priority types, with the same total messages transmitted and average message sizes for each of these three priorities. However, the very uneven load nature of the trace (as we have seen in the *"kp176half.dat"* trace plot earlier) means that the stations provide very different proportions of the total load. Therefore, for PG based systems, regardless of the intra-station shuffling status, it was only possible to produce equal loads by choosing three station subsets with very disparate (i) number of member stations per subset, (ii) average message sizes produced by the subset and (iii) total messages generated by the subset. This is evident when one compares the left and right hand sides of Figure 6.20.

Rather than a single factor such as overall traffic load, it is this plus all three factors (i) through (iii), as well as (iv) the traffic stream autocorrelation level within individual stations, which jointly account for the overall traffic profile of each priority presented to the IM server. With reference to the 10 station IM

system delay curves, Figure 6.16 through Figure 6.18, we note that the two PG systems (unshuffled and shuffled trace) retain the same performance relative to each other for all three priority types, because they consist of three identical station subsets in terms of the above mentioned factors (i) - (iv). On the other hand, the RMP system has different levels of similarity to the PG systems in terms of the factors (i) - (iv), for each individual priority, causing differences in the overall traffic arrival profile on a per-priority level, and therefore affecting the queueing behaviour of the messages. This is manifested in Figure 6.16 through Figure 6.18 as a different relative delay performance between the RMP and PG system curves for each individual priority level.

6.3.1.3 The Impact of F-CPR's Go/Stop Effect on 10 station PG and RMP Systems

An RMP system where all stations can and do transmit messages of any priority (randomly and evenly assigned on a per message basis), can be thought of as "chopping" up any high-correlation artefacts (i.e. large bursts) of the original unshuffled real traffic trace. In effect, all incoming messages are on average assigned a given priority every three message arrivals; this clearly causes longer interarrival times in a particular station for each priority, than those in the PG systems where a station will **only** have arriving messages of one priority. It is thus highly unlikely in an RMP system, to have occurrences in the aggregate system-wide arrival profile of just one or two stations transmitting a number of consecutive same-priority messages for a sustained period of time. This is something which is more likely to occur in an unshuffled-trace PG system, where if one same-priority station happens to be exclusively active for any given period of time, then the Head-End server will only see messages of that particular priority. Note that a shuffled-trace PG system has most of the correlation destroyed anyway, which is why F-CPR matches IM performance so well, at all priorities and both system sizes, for such a shuffled-trace PG system (see Figure 6.16 through Figure 6.18). We observe that the F-CPR / IM matching is about equal for the RMP and shuffled-trace PG systems.

In essence, eliminating stream correlation in one way or another (RMP or PG with shuffled trace) causes the delay performance of either the F-CPR or IM to become *immune from protocol artefacts* and to be solely determined by the queueing behaviour of the aggregated traffic profile at the Head-End. Something to learn from this observation is that testing the protocol with PG systems, regardless of the priority scheduling scheme employed, but with the trace being unshuffled (i.e. in its original state), is a more stringent examination because it takes into account both normal queueing and protocol-specific effects. The best example of this is to be seen in the set of medium priority curves in Figure 6.17 where, although the IM benchmark shows the RMP system to suffer worse performance from a queueing-only standpoint than the unshuffled-trace PG system, the throughput-limitation characteristic of F-CPR is significant enough to reverse this observation in both the JET and SA schemes. This causes clearly worse performance for the F-CPR unshuffled-trace PG system. In the case of the go/stop affected F-CPR PG systems, it is the medium priority traffic which is more adversely affected (compared to IM performance) than the low priority traffic. This occurs because of differences in the particular traffic arrival profiles of these two priorities (i.e. one profile is more susceptible to the go/stop phenomenon than the other.

6.3.2 Benchmarking F-CPR:JET against the IM - High Priority Traffic Delay Performance

Figure 6.21 shows all the signs necessary to declare that the F-CPR:JET implementation stacks up very well against its IM benchmark. That is, all IM and F-CPR:JET high priority traffic access delay curves shown are almost identical in shape and separated by an approximately constant value equal to the size of the unavoidable *RTD* overhead, under three diverse conditions - wide ranging loads; different number of stations in the system; and, input trace processes with different priority assignment mechanisms and hence, different intra-station correlation structures.

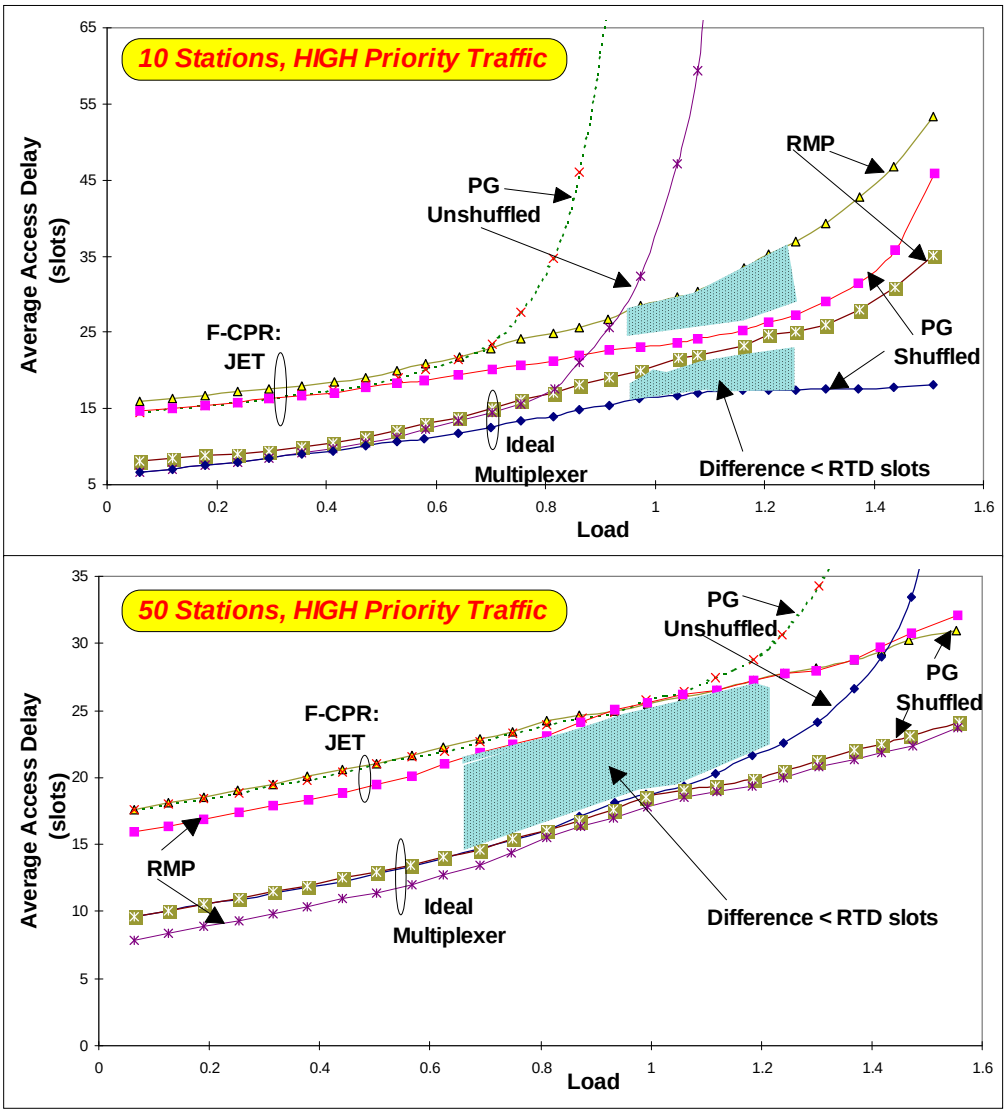


Figure 6.21: Benchmarking the F-CPR : JET Protocol against the IM

It has been emphasised that the delay curve separation is **approximately** equal to the *RTD* value, because in Figure 6.21 there are load regions (as marked) in both the 10 station and 50 station systems of either type (PG or RMP), where the delay difference drops below *RTD slots*! Although even in these special regions the difference between the curves is not very much lower than *RTD* slots (e.g. for the system under consideration, *RTD* = 8 slots and the lowest difference is 6.6 slots), the simple fact that it **is** lower by any

amount may at first glance seem to be impossible since we stated at the beginning that the *RTD* overhead is something which F-CPR must implement in order to enable proper functioning of its signalling and reservation functions. The answer to this apparent mystery lies in the **pre-emptive** nature of the JET priority scheduling scheme. Although each message sent by the protocol is still subjected to an extra *RTD* slots of access delay as compared to those sent by the IM, the pre-emptive nature of the JET scheme gives better queueing performance to the high priority messages than the non-preemptive multiple priority IM we have used as a benchmark.

As Figure 6.21 shows, JET treats high priority better than the IM in all *intra*-station correlation scenarios; in addition to this observation, it has also been found that the JET scheme exhibits superior high priority traffic queueing performance to the IM, in cases where the *inter*-station correlation is at a maximum. Table 6.3 provides an example of such a case. The data in this table was taken from the tests carried out to investigate the impact of inter-station correlation (presented in Section 6.4.2); a sample load point (0.4409) is shown, with both the F-CPR:JET and IM systems using the deterministic, maximally inter-station cross-correlated trace read process, which will be described in Section 6.4.2. The table reinforces our observations from Figure 6.21, showing that with the JET scheme in place, the medium and low priority traffic suffer a slightly worse than *RTD* access delay difference compared to the IM, to the benefit of the high priority messages whose average delay is less than that of the IM by an almost equal amount.

Priority	***Ideal Multiplexer Average Access Delay (slots)***	***F-CPR:JET Average Access Delay (slots)***	***F-CPR - IM Difference (slots)***
High	*14.238*	*21.010*	*+6.772*
Medium	*17.404*	*25.896*	*+8.492*
Low	*18.725*	*28.062*	*+9.337*
Total	***16.827***	***25.050***	***+8.223***

Table 6.3: IM vs. F-CPR: JET delay comparison for a 50 station, Unshuffled-trace PG System at a sample Load = 0. 4409

Let us now illustrate the pre-emptive nature of JET by contrasting its scheduling algorithm to that of SA, with a simple example shown in Figure 6.22 (overleaf). The setting of the example is as follows: we have a single high priority station currently emptying its queue using reservation messages. The *RTD* is 8 slots, with the one-way coaxial propagation delay (PD) being 1 slot. From Figure 6.22 we note the existence of so-called “windows of opportunity” - periods of time equal to *RTD* = 8 slots, in between the consecutive high priority messages. These windows provide an opportunity for messages of lower priority (medium / low) to efficiently fill the periodic inter-high-priority-message gaps. The first high priority message request has been drawn at slot 1, with its following message scheduled for arrival in slots 10, 11 and 12.

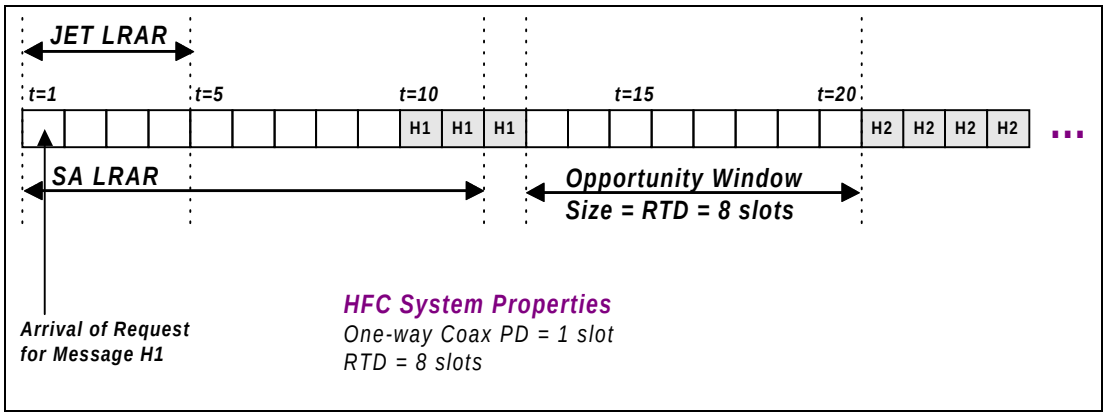


Figure 6.22: Low-Priority Request Arrival Regions (LRARs) for the SA and JET Schemes

Also highlighted in the diagram are two overlapping *low-priority request arrival regions* (LRARs): (i) the JET LRAR and (ii) the SA LRAR. The significance of these regions is that any lower-priority request arriving within the *"X" LRAR*, is scheduled by a Head-End running the *"X"* priority scheduling scheme *before the final cell of the first high-priority message arrives at the Head-End and has the chance to make its usual periodic reservation*. This means that a low-priority request arrival within an LRAR has the potential to violate the size of the window of opportunity (e.g. with a 30 cell request) and hence disrupt the access delay of the next high priority message. If a low-priority request were to arrive at timeslot 7, it would fall into the SA LRAR but not the LRAR, and would only be scheduled by a Head-End running an SA algorithm. Therefore, the smaller the LRAR associated with a scheme, the better the level of priority afforded to traffic nominally assigned the label "high priority". From Figure 6.22 we can derive equations (6.1) and (6.2) and show that the JET LRAR is a subset of the SA LRAR:

$$LRAR_{SA}(M_i) = S(M_i) + RTD \tag{6.1}$$

$$LRAR_{JET}(M_i) = S(M_i) + 1 \tag{6.2}$$

where M_i refers to high-priority message number i , and $S(M_i)$ is the message size in cells. The notation $LRAR_{JET}(M_i)$ stands for the size in slots of the low-priority request arrival region associated with message number i, for the JET scheme.

The reason why the testing is based on comparing a pre-emptive scheme such as JET with a non-preemptive Ideal Multiplexer is because the other priority scheduling scheme which we are testing, SA, is also non-preemptive. The logic of this kind of comparison revolves around first establishing a performance benchmark for the existing SA scheme against the IM, and then seeing how the key performance indicators such as delay and utilisation are changed for the various traffic priorities, when the JET scheme is implemented. We have shown here that in some relatively infrequent instances (but importantly under overload conditions) for both system sizes, the high priority traffic enjoys better queueing performance when it is treated by the JET scheme as opposed to when it is handled by a non-preemptive IM. And as expected under the work-conservation principle, the better delay performance of higher priority traffic provided by JET (as compared to that provided by say a non-preemptive IM) is in a

way subsidised by the equivalently worse performance of the traffic with lower priorities. The exact amount by which each of the lower priority delays are worsened largely depends on the system type (PG or RMP) and the spread / correlation trace characteristics, as discussed in Section 6.3.3.

6.3.3 Effect of Number of Users for an Unshuffled PG System

As was highlighted in Section 5.2.5.3, the ***unshuffled PG system*** is seen to be closest to a realistic snapshot of typical HFC system activity, and thus most of the tests carried out are based on this particular combination of real traffic trace and priority assignment mechanism. Figure 6.23, depicted on the following page, illustrates the effect of the number of stations in the system, on the average access delay. All three systems (F-CPR:JET, F-CPR:SA and IM) are shown, and the figure has a graph for each of the priorities.

The first observation is that, like in the single priority systems, the delay performance of all three 10 station systems is significantly worse than that of the 50 station systems - for each of the three priorities. Recall that it was found that when a real trace is used as input to a larger system instead of a smaller one, the dual effects of traffic smoothing and load balancing improved the performance.

The second observation we can make is that the proximity of the F-CPR delay curves to the IM curves varied on a per-priority basis with a strong relationship to the priority level, in the case of the 10 station system but not the 50 station system. This is once again a side-effect of the F-CPR's throughput limitation, and hence it is not unexpected that the 10 station system is significantly more adversely affected than the larger 50 station system (for reasons made clear earlier). With this throughput-limitation a known and explored quantity, we now become primarily concerned with the effect of priority levels implemented in a protocol susceptible to RTD-based inefficiency - that is, how does priority impact the effects of the go/stop phenomenon?

Focusing on the F-CPR : SA delay curves for the 10 station system shown in Figure 6.23, it is apparent that the higher the priority of the traffic, the worse the mismatch with the IM curve, or in other words, the more prominent the effect of the go/stop phenomenon. Although the logic behind this observation may not be at first obvious, it makes sense when the behaviour of any priority queue is taken into account. In a non-preemptive multiple priority queueing system traffic of a given priority neither *sees* nor is *affected by* any messages of lower priority waiting in their respective buffers (or in a joint buffer for that matter), except for a lower priority message that might be finishing its service time upon the higher priority message's arrival. Because the latter will occur rarely, it can be stated that any real extra delay incurred for messages of a given priority will come primarily as a result of queueing behind and waiting for all messages of a higher priority to be served.

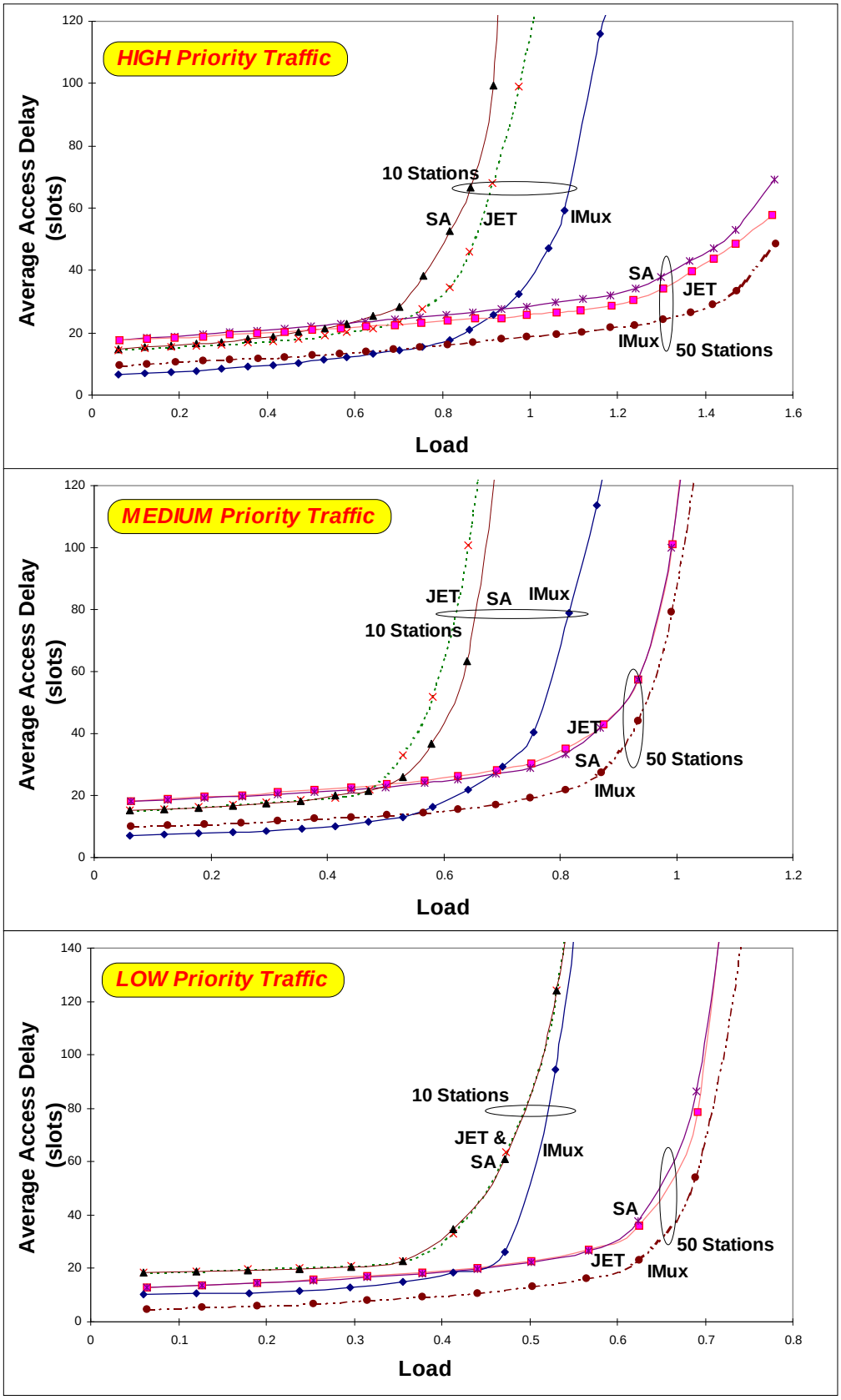


Figure 6.23: Effect of Number of Stations (10 vs 50) for the Unshuffled Trace PG System

Therefore, with reference to the Figure 6.23 F-CPR : SA curves, the highest priority traffic behaves (almost) as if it has sole use of the channel, while at the other extreme, the lowest priority traffic will at almost all loads have as a significant portion of its overall average access delay, the time spent waiting behind all the higher priority messages. The implications of this with the go/stop phenomenon are that the

active high priority stations will, from their point of view, suffer almost the full impact of the go/stop phenomenon, as if we were dealing with a single priority system; the medium and low priority stations will have the impact of the go/stop phenomenon masked to an extent by waiting behind the higher priorities, and the lower the priority - the greater this "mask", explaining the narrowing of the F-CPR - IM delay curve difference with lower priority in Figure 6.23.

It has been found that the masking effect just described, heavily depends on the traffic arrival profiles for the three priority classes. As an example to aid understanding, we now focus on an example system, where the masking effect is quite noticeable since it is a PG system which (out of 10 stations) has an allocated subset of only 3 high priority stations each producing a traffic stream of high correlation at different times. Thus, at any time when only one of these stations is active, it is *seen* by the system to be **almost** the only transmitter on the channel. The statement emphasises "almost", because of the nature of the SA scheme. During a period of time when there is a single high priority station (with $RTD = 8$) being served by a Head-End employing SA, there are *windows of opportunity* of duration $RTD + 1 = 9$ slots during which an arriving lower priority request may be scheduled in between consecutive high priority messages, without affecting the high priority traffic performance at all.

That is, if a lower priority message is of size 8 cells or less, then it may be neatly packed into the available gap without affecting the high priority messages at all. Considering that for the 10 station PG system, the average message sizes for the medium and low priority messages are 4.89 cells and 11.19 cells respectively and the opportunity window size is 8 slots (see Figure 6.20 and Figure 6.22), it is evident that even if messages of a lower priority do arrive to be scheduled during such windows of opportunity, their overall effect on high priority traffic scheduling is minimal. These windows of opportunity (whose size is solely dependent upon the *RTD*) exist in both the SA and JET schemes, as quantified in the previous section.

It should be clear that while the probability that the windows of opportunity are violated by a larger low priority message (which does not "neatly" fit within *RTD* slots) is generally small for both priority scheduling schemes, it is much smaller for the JET scheme due to its pre-emptive nature. As explained earlier in Section 6.3.2, the pre-emptive service discipline of JET dictates that the lower priority messages are not to be scheduled until the last possible timeslot. This paradigm is illustrated by the example earlier given in Figure 6.22, from which one can conclude that in the presence of a long train of consecutively scheduled high priority messages, the probability of a lower priority being scheduled such that it potentially violates the size of the opportunity window, is greatly reduced when we are dealing with the JET scheme. This reduction is 60% in the example, but depends on factors such as average message size and *RTD*. This greater level of channel domination afforded to high priority messages by JET manifests itself in Figure 6.23 as a better high priority F-CPR - IM matching than for the corresponding SA curves, at the expense of an equally worse medium priority F-CPR - IM matching than for the corresponding SA curves. Interestingly, the effect is not nearly as pronounced when one looks at these same curves for the 50 station system; with the larger system, and smoother, more evenly distributed traffic, the throughput limitation ceases to be a problem and the two priority schemes converge to near identical performance. Of

course, the pre-emptive nature of JET will mean that it will always give slightly better delay performance to high priority traffic, even though this may be insignificant.

In both the 10 station and 50 station F-CPR: JET PG systems, the reason that it was the medium and not the low priority traffic which suffered more due to the better high priority traffic performance, is that the low priority requests must spend a larger proportion of their time waiting for both of the two higher priorities anyway, so they are less affected by any changes to the scheduling order of the two higher priority messages. Furthermore, the arrival profile of the different priorities of traffic was such that there were no long periods of high and low priority message arrivals only, without any medium priority message arrivals. If such a scenario had happened, then the low priority message delay would no doubt have been more significantly worsened by the JET pre-emption feature.

6.3.4 Performance of Scheduling Schemes per Priority for an Unshuffled PG System

Figure 6.24 and Figure 6.25 shown on the following two pages, highlight the delay and utilisation performance of all traffic priorities of the IM, F-CPR:JET and F-CPR:SA systems with 10 and 50 stations respectively. Once again, we chose the PG priority assignment mechanism for this investigation. All of the issues associated with delay performance of the JET and SA schemes have been explored in previous sections, so we will not go into any detail here. However, it is prudent to once again briefly focus on the better high priority traffic delay performance of the JET scheme compared to the SA scheme, and to mention that this is more pronounced in the smaller sized system, when the protocol's throughout-limitation poses more of a problem. Recall that earlier it was found that the superior delay performance also holds regardless of the intra-station correlation level (i.e. the priority assignment mechanism adopted), and is due to the JET scheme's pre-emptive approach to scheduling, as explained in more detail in Section 6.3.2.

Let us now focus on the utilisation performance of the F-CPR and IM systems - the bottom halves of Figure 6.24 and Figure 6.25. As expected, one can observe that the pre-emptive nature of the JET scheme allows it to reserve a greater proportion of the channel for its high priority messages, than that possible by its non-preemptive SA counterpart. As was the case with the delay curves, the impact of JET's high priority bias is more pronounced in the smaller system (Figure 6.24) because of the increased influence of the go/stop phenomenon with only 10 transmitting stations. The JET's tendency to allow higher utilisation to high priority traffic comes to the fore under conditions which exacerbate the go/stop effect (i.e. single high priority transmitter), because, it is under these conditions that the differences in the JET and SA scheduling disciplines are highlighted, as was shown in Section 6.3.2 (the diagram in Figure 6.22, in particular).

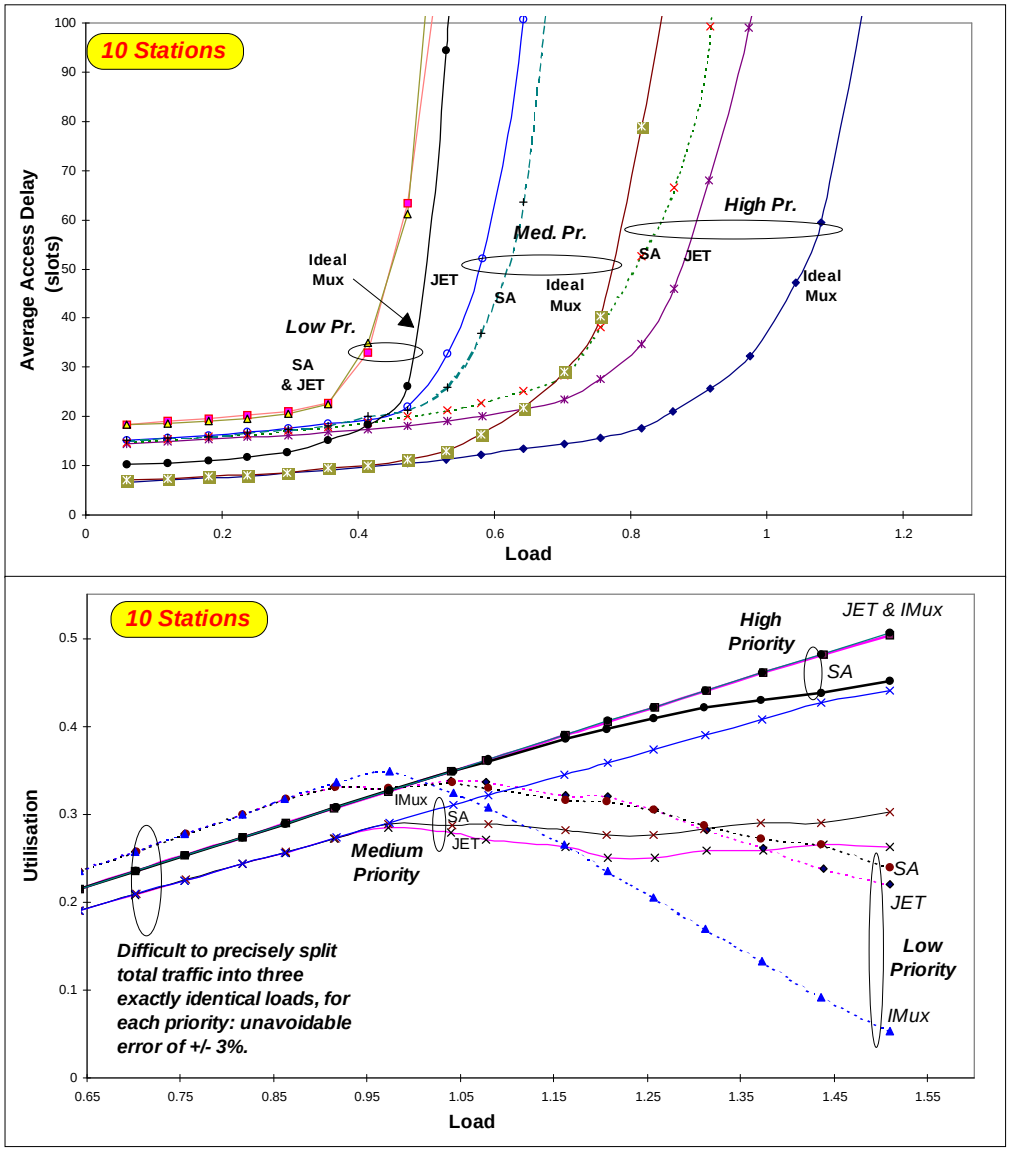


Figure 6.24: Delay and Utilisation Performance of the Three Priority Levels

10 Station Unshuffled PG System

Another interesting observation to be made from the utilisation curves in both Figure 6.24 and Figure 6.25 is the impossibility of exactly balancing the offered load in a PG system, measured in total generated cells (not messages). This results in utilisation vs. load curves which do not exactly overlap. Also, note in Figure 6.24 how the medium priority traffic behaves dramatically differently when carried by the F-CPR (SA or JET) as opposed to when handled by the IM. The markedly reduced utilisation we see for medium priority traffic comes as a result of the impact of the go/stop phenomenon on traffic of the medium priority class, for this small system (only 10 stations). That is, the particular arrival profile of the medium priority traffic is such that long periods of single-station activity occur, non-concurrently with high priority traffic activity, meaning that there are long periods of simulation time when medium priority messages see no other "enemy" to their throughput performance, other than the fixed *RTD* overhead. The particular scheduling scheme does not play a big role here, and so the result is the wildly different IM - F-CPR medium priority curves shown in Figure 6.24. In situations like this, the low priority traffic happily fills all the gaps left to it by the medium priority traffic, and hence "soaks up" much more utilisation than would be apportioned to it by a multiple priority IM (see the bottom part of Figure 6.24).

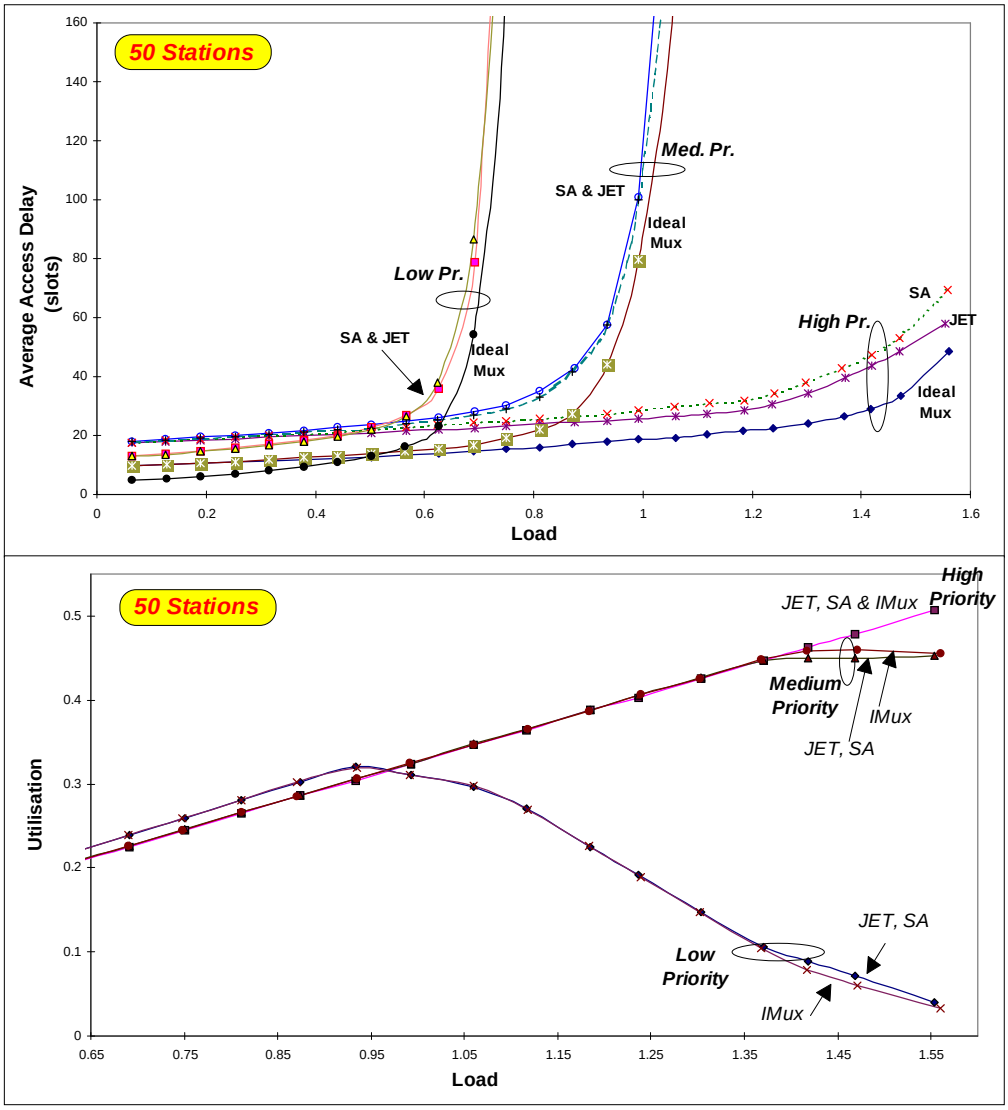


Figure 6.25: Delay and Utilisation Performance of the Three Priority Levels

50 Station Unshuffled PG System

A comment needs to be made about the overall utilisation performance of the two schemes (JET and SA). Namely, regardless of the way in which these two schemes split the available channel capacity among the three priority levels, the overall level of utilisation that each of the schemes achieves is identical. This is arguably intuitive, because although the schemes may differently arrange the scheduling of the priorities on the upstream channel, something is always placed on the channel by either scheme and no wastage occurs. Look at the bottom half of Figure 6.24 for example; the medium and low priority traffic enjoys better utilisation if handled by the SA scheme instead of the JET scheme - the sum of this benefit is about 0.06. On the other hand, the high priority traffic handled by SA receives a portion of the utilisation "pie" smaller than high priority traffic handled by JET. The amount by which this pie is smaller is also 0.06, and exactly balances the benefit to the other two priority classes. Although it was not significant enough to graph, our investigation also found that the total channel utilisation of the F-CPR multiple priority systems, whether they be using JET or SA, was just slightly improved over the single priority F-CPR system utilisation (slightly better improvement was noticed in the case of the 50 stations system). The same is not true of the IM (it cannot hold true theoretically). The key to this observation lies in the

different way of scheduling introduced for a multiple priority system. Namely, instead of scheduling all traffic immediately as it arrives, the lower priority traffic is placed in holding queues and waits for slots during which there is an absence of high priority request arrivals. The likely presence of at least one lower priority message in one of the holding queues at most medium to heavy loads, slightly increases the probability of efficiently packing (a portion of) any gap in the upstream channel, which would otherwise pass upstream as unfilled in a single priority system.

6.4 The Effects of Intra- and Inter-station Correlation

6.4.1 Impact of Intra-station Traffic Correlation

The focus of this section is the impact on overall F-CPR performance, of load intensive bursts of long duration arriving within a single station's queue. In this case, we are unconcerned with the cross-correlation effects between various queues - this will be the topic of the next section. In order to achieve a comparison of extremes, we test the IM and F-CPR protocol with two versions of each of the two traces *"kp176.dat" and "kp176half.dat"*, one shuffled and the other one unshuffled (as was originally recorded). The exact mechanics of the shuffling process have already been explained earlier in Section 2.6.2 - in short, the process is designed to retain exactly the same load per station, distribution of load among stations and average message size, while randomly shuffling up the original recorded trace. In this way we destroy any trace of self-similarity and correlation within each individual station's arrival process, as has already been demonstrated by measurement of the Hurst parameter for each of the four trace combinations (shuffled/unshuffled and half/full version of the trace).

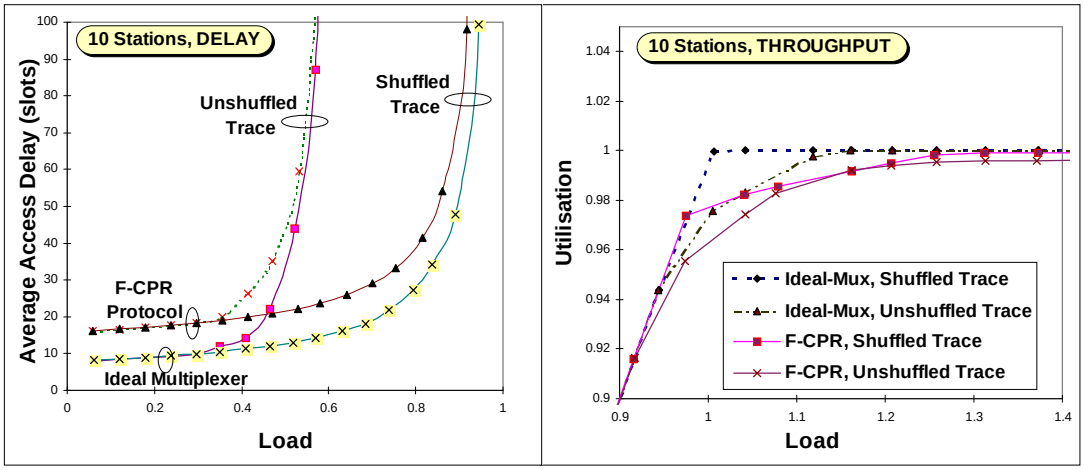


Figure 6.26: F-CPR and IM Delay and Throughput, 10 Station System

The IM and F-CPR could have been tested for both 10 and 50 station system sizes with the shuffled and unshuffled versions of the trace in its original form, *"kp176.dat"*. However, it was chosen to use the halved version of the trace, *"kp176half.dat"*, when testing the 10 stations system. The reason behind this was the desire to find out the effect of halving the trace, on F-CPR performance (which in the case of the 10 stations system, was already severely affected by the go/stop phenomenon explained earlier in detail). This effect becomes clearly visible when one compares Figure 6.26 (previous page) to Figure 6.5 and

Figure 6.7. From both the delay and utilisation point of view, when the halved instead of the full version of the unshuffled trace is used for a 10 station system, the F-CPR protocol performance is superior and as such it is much closer to that of an Ideal Multiplexer. Unlike in Figure 6.7 where the two IM and F-CPR curves begin to diverge, Figure 6.26 shows that the unshuffled-trace F-CPR delay converges to the IM delay quite rapidly after a load of 0.5. In addition, Figure 6.26 shows that the utilisation of bandwidth at the load of 1.0, is far better at 0.96 than it was in Figure 6.5 at 0.84. Note that the utilisation and delay performance of the IM remain unaffected, regardless of whether the half or full unshuffled trace version is used.

This superior performance of the F-CPR, but not of the IM, due to the running of a halved version of the original recorded (unshuffled) trace is directly linked to how the properties of these two versions compare to each other. Namely, it was mentioned in Section 2.6.4 that by halving the original trace, most of the zero readings are eliminated, and a more even activity level is achieved (not without variations of course, but largely non-zero). This has the effect of more evenly spreading the load among all stations, as evidenced by superior F-CPR protocol performance due to the easing of the go/stop effect limitations (recall that better performance is achieved with a traffic profile generated by many transmitters rather than by one or a few). As mentioned earlier, the IM performance is not affected at all, and this is because halving the trace does not reduce in any significant manner the correlation and self-similarity of the individual stations' traffic streams.

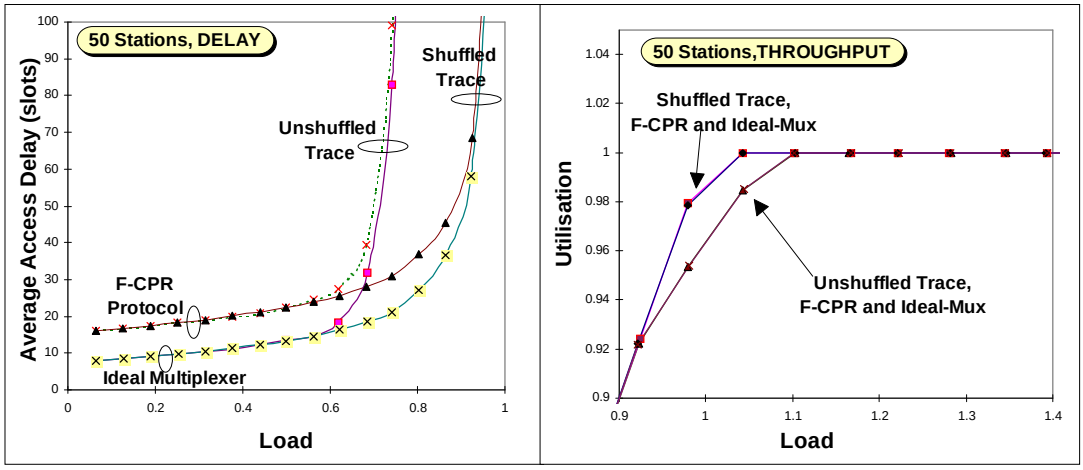


Figure 6.27: F-CPR and IM Delay and Throughput, 50 Station System

Now we shift our focus from just the unshuffled trace loading the 10 station system, to look at the more globally pervasive effect of shuffling either trace version (half or full) for either of the two system sizes. Interestingly, at this more global level, it is solely the queueing behaviour as dictated in the IM, which affects delay and utilisation performance, and not the F-CPR's intrinsic artefacts and limitations. A single server with multiple queues will gain in performance both from a utilisation and delay point of view (see Figure 6.26 and Figure 6.27), if the input traffic becomes less correlated [ZUKE 86]. From the delay graph pairs in these figures, we note that the number of these multiple queues (read stations) is a factor - the performance benefit of using a shuffled trace is more significant with a smaller system. This is because, when using the unshuffled trace, the smaller system retains more of the original trace's

correlation structure and hence will perform more badly than a large system using the unshuffled trace. A shuffled trace on the other hand has all of the correlation structure destroyed anyway, and becomes equivalent to a memoryless process, thus the system size becomes totally unimportant.

A final important fact to note from Figure 6.26 and Figure 6.27 is that at low loads, the delay performance of both the unshuffled and shuffled traces, regardless of system size, is completely identical - correlation and self-similarity artefacts of individual traces are at this point still too far "spread" by the inserted silence periods (for a description of the trace read process mechanics, refer back to Section 2.6.2). Then, at some *threshold value* (which is larger for the inherently less-correlated 50 station system traffic profile, as just explained above) the shuffled and unshuffled curves part, with the latter increasing in delay rapidly to infinity. This threshold corresponds to a point where one or more stations undergo a long-enough period in which the average interarrival period of the messages is shorter than the average service time capability of the server, to cause a large enough *backlog* in one or more of these queues. After this point, because the server is fair and cannot clear out entirely the backlogged queue(s) (even though there may be only one or two), the load grows further and more station queues start growing in an unstable manner, and, the queues which were affected first continue to get worse, in a so-called *avalanche effect.*

6.4.2 Impact of Inter-station Traffic Correlation

Unlike in Section 6.4.1, here we look at a more global and system-wide effect of correlation. That is, rather than being concerned with how intensive and long bursts arriving within a **single station's** queue affect overall F-CPR performance, we look at the impact of bursts of traffic simultaneously arriving from **many stations**. As always, the extreme case is of interest and has been studied; and that is the case when the real trace simulation process is purposely made to only allow the stations to register arrivals at certain designated instants in time. This is not to say that all stations will always register an arrival at one of these designated instants, but rather that if they do, it will necessarily be at one of these predefined batch arrival slots. This means that some of the readings, as shown in Figure 6.28, will sometimes be zero.

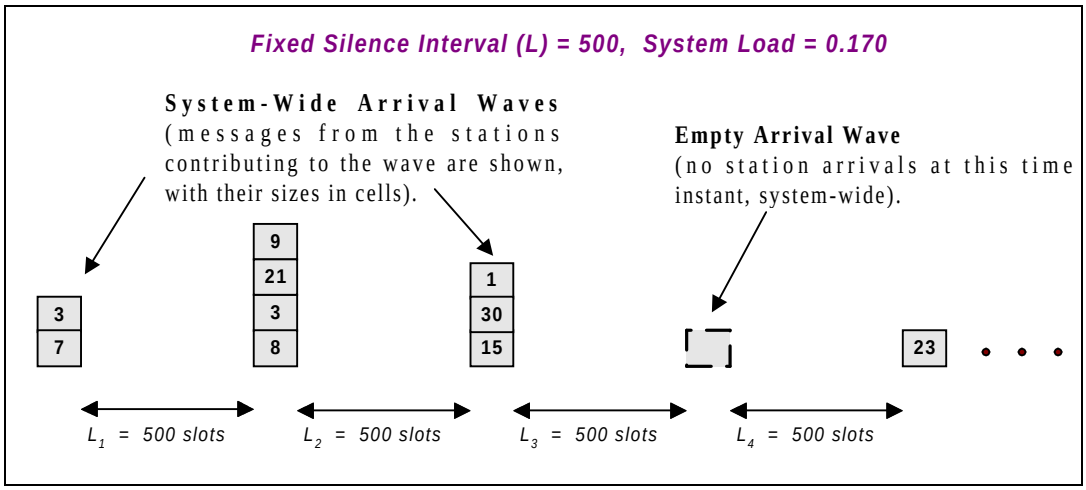


Figure 6.28: Maximum Inter-station Correlation Effect

The net result of this type of trace read process is that from the point of view of the Head-End, it is hit with periodic arrival waves, made up of a subset of stations, which can vary in size from no stations to all stations. No other arrivals can be registered outside of these periodic arrival waves. This trace read

arrangement represents the worst-case scenario in terms of inter-station correlation, and is the one we chose to study.

6.4.2.1 Single Priority System

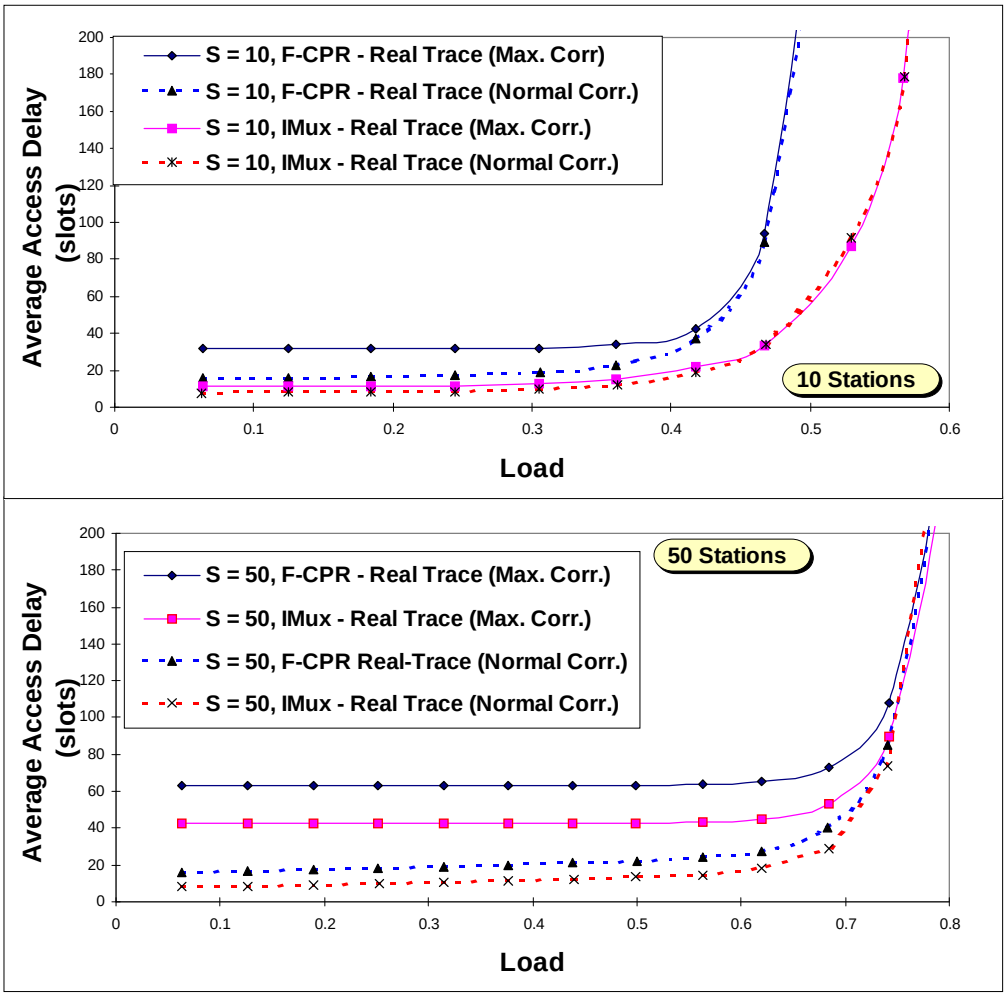


Figure 6.29: Effect of Inter-station Correlation on Average Access Delay

Figure 6.29 contrasts the average access delay curves for HFC and Ideal Multiplexer (IM) systems with a small (10) and moderately large (50) number of stations, under two scenarios, purposely chosen to have diametrically opposite inter-station correlation properties. The definitions of the two scenarios are:

(a) Normal Correlation Scenario: refers to the original trace read process described in Section 2.6.2, whereby the global system arrival process may record any number of arrivals in any slot. Any coincidental arrivals of messages from two or more stations within a single slot are determined purely by chance.

(b) Maximum Correlation Scenario: refers to the maximally correlated inter-station read process shown in Figure 6.28 and described above, whereby no message arrivals can be registered outside of certain designated slots which are hit by multiple-station message arrival waves.

From Figure 6.29 three important observations can be immediately made, when comparing the delay performance of the Normal Correlation curves to the Maximum Correlation curves: (i) the Ideal

Multiplexer is less affected than the F-CPR protocol by the inter-station correlation; (ii) both the IM and F-CPR protocol are more detrimentally affected by inter-station correlation associated with a larger number of active stations; and, (iii) for both the IM and F-CPR protocol curves, the inter-station correlation effects are only visible prior to the onset of congestion threshold, described in Section 6.4.1 - the value of this threshold load and the shape of all curves after it is exceeded both remain unaffected by inter-station correlation.

These observations may be explained by the differences between the IM and F-CPR, and by queueing theory fundamentals. Observation (i) arises due to the fact that the IM has full state knowledge about each of the stations' queues, and thus does not need a contention-based signalling channel. F-CPR will clearly suffer greatly worse delay than the IM, because as well as experiencing the normal queueing delay penalty associated with large batch arrivals, it has to resolve multiple contentions from the stations' simultaneously arriving messages, at each periodic arrival wave slot. Observation (ii) is related to the batch size which "hits the server" at each periodic arrival wave time instant. That is, larger batch sizes will, on average incur a greater service-time related waiting delay while the member messages of the batch are served one by one. It stands to reason that an individual message within a larger batch will spend more of its queueing delay waiting for the service of the messages which had arrived with it, to be over.

Observation (iii) is based on the fact that simultaneous (i.e. batch) arrivals from many stations will have a significant impact when all the queues system-wide are near-empty or lightly loaded, because then the worse delay observed is solely a function of the batch population, arrival frequency and average message sizes within the batch. This means that the inter-station correlation is only a factor before the onset of congestion threshold is reached, because after this point, the degradation in queueing performance is dominated not by the problems associated with batch arrivals, but by the unstable backlogs leading to the system-wide avalanche effect, as described in Section 6.4.1. As stated in that section, the actual value of the threshold is a function only of the *intra*-station traffic correlation.

A final note of interest is that the throughput (overall system utilisation) performance of either the F-CPR or IM was found to be only slightly negatively affected by inter-station correlation. It is perhaps intuitive that the IM only suffers delay and not throughput degradation, since signalling is never required and usable slots are never wasted. However, ordinarily, one would expect the increased probability of collisions under inter-station correlation to cause some amount of bandwidth wastage for the F-CPR protocol, which in part relies on collision-prone (CMS) signalling. The realistic traffic trace we used is such that only a small number of heavily loaded stations are active for long periods of time, meaning that their queues are usually non-empty and the piggybacked DMS signalling feature of F-CPR may be employed. This explains why, for the trace we examined, it is only the bursty arrival *patterns within individual stations' queues (i.e. the intra-station correlation)*, that can then lead to long silence periods and cause some level of utilisation wastage.

6.4.2.2 Multiple Priority System

Our investigation of the inter-station correlation effect on a multiple priority system yielded for the most part, identical observations to those made when the single priority system was under investigation, both regarding delay and utilisation performance. The only noteworthy finding which may be said to be unique to multiple priority systems is the change to the high priority traffic average access delay curves introduced by maximally cross-correlating the station traffic. The top part of Figure 6.30 shows the high priority traffic delay curves for a 50 station system with normal inter-station correlation (the same definitions to normal and maximal correlation apply, as given in Section 6.4.2.1), while the bottom part shows how these curves are affected when the inter-station correlation is at its maximum level.

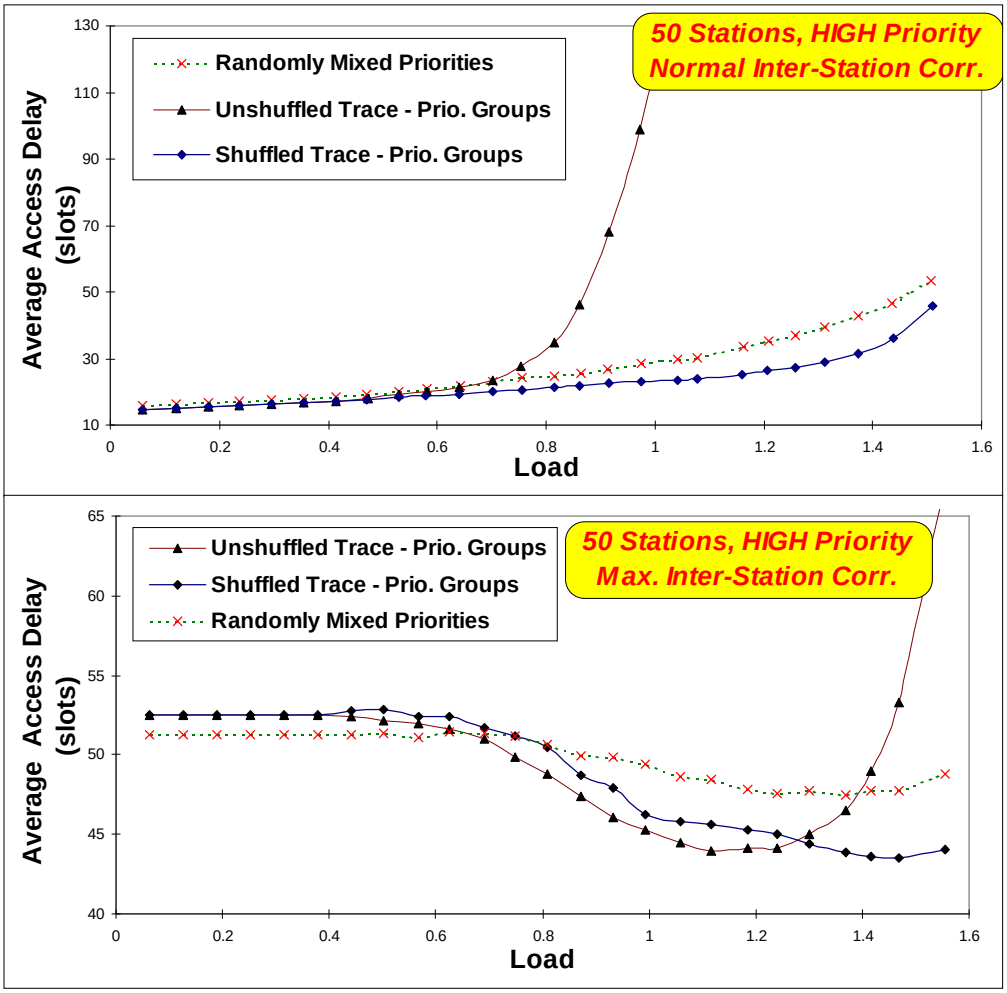


Figure 6.30: Inter-station Correlation Impact on a Multi-Priority System

The average message size of the RMP 50 station system, using the *"kp176.dat"* trace, is 5.96 cells/message for all priority types. However, the shuffled and unshuffled PG systems have priority-dependent message sizes, and in the case of high priority the value is 5.27 cells/message. This is why, in the top part of Figure 6.30, when there is no cross-correlation between stations, cells in the RMP system have a slightly longer waiting time and hence the system has higher average access delay than the other curves at low loads. At a load of about 0.5 the three curves begin to diverge more significantly and the shape of the graph and final relative positions of the curves are determined by the intra-station correlation

effects for high priority traffic, with the unshuffled PG system being the worst performer, as expected. It is followed by the RMP system which is considerably better thanks to its "chopping-up" the station traffic into three different priorities, thereby destroying some of its inherent correlation. Of course, the best delay curve is that of the shuffled PG system where all correlation artefacts have been destroyed.

We now focus on how this more-or-less intuitive set of curves, in the top part of the figure, is radically changed both in relative positioning and absolute values, when maximal cross-correlation between the stations is introduced (shown as the bottom part of Figure 6.30). Four principal observations can be made: (i) all delay curves are much higher at low to medium loads, than in the top part of the figure, (ii) at low to medium loads (0 - 0.55), the RMP system enjoys a slightly better delay than that of the two PG systems, which are identical in this range, (iii) after a load of 0.5, all three delay curves actually begin decreasing with higher load, with the unshuffled PG system having the highest rate of decrease and the RMP system the lowest; and (iv) while the shuffled PG and RMP systems experience delays very close to those without the cross-correlation under overload, the unshuffled PG system greatly benefits from the cross-correlation and rises to infinity at a load of 1.5 instead of 1.0.

In order to address (i), we recall a similar observation which was made in case of the single priority system, where it was explained that the F-CPR will clearly suffer greatly worse delay than the IM, because apart from experiencing the normal queueing delay penalty associated with large batch arrivals, it has to resolve multiple contentions from these batches which are made up of messages from different stations at each periodic arrival wave slot. This same logic applies to a multiple priority system, and the average access delay curves of all priorities are found to be equally increased by inter-station correlation at low loads; this happens because at low loads, when the contention-based signalling delay (as opposed to Head-End queueing delay) dominates, the CMS minislots are not treated by the CMS signalling channel's contention resolution algorithm preferentially (according to priority level). Observation (ii) is accounted for by the trace characteristics and the mechanics of the trace read process for the RMP and PG systems. Close examination of Figure 6.9 in Section 6.2.3 shows that the *"kp176.dat"* trace, when used for testing the 50 station system, results in approximately 13 very active stations (i.e. those providing virtually all of the system load during the simulation). Using this figure of 13, the message priority assignment method for a batch-arrival RMP system, dictates that on the average in each arrival wave 13 / 3 = 4.333 stations will generate messages of high priority (i.e. this is the batch size for RMP systems). On the other hand, the PG systems require an assignment of stations to three subsets, one for each priority, balanced in such a way that each subset of stations generates an approximately equal cell load. By a random choice of subset membership, both PG systems have a high priority subset of 8 stations, 6 of which can be said to fall into the category of very active stations (thus giving a batch size of 6 for PG systems). Therefore, it is no surprise that at low to medium loads, when the batch size determines queueing behaviour as well as the impact on F-CPR's signalling channel, the RMP system will enjoy a slightly better delay performance than the two larger-batch-size PG systems.

Referring to observation (iii), as a general comment applicable to all three systems shown in Figure 6.30, (RMP, PG unshuffled and PG shuffled), it can be stated that the reason for the overall decrease of average access delay with increasing load, is that at low loads when the batch arrival effect dominates, all the

priorities are treated equally because the contention resolution system does not discriminate; then, as the load increases the high priority traffic is shielded by its foremost priority status, from the negative effect of scheduling-related queueing (at the Head-End), while simultaneously enjoying the benefit of increased piggyback DMS signalling. In this way, as the high priority queues begin to fill with higher system-wide load, the collision factor is slowly removed and the importance of priority labels begins to again emerge. In Figure 6.30 we have only shown the results for a 50 station system, but it was found that the increased rate of CMS collisions in this larger system serves only to make the effect more pronounced, as opposed to that noted in the 10 station system.

Furthermore, of the two PG systems, it is the one with the more highly intra-station correlated (i.e. unshuffled) trace which reaps greater benefits from the usage of DMS slots, due to its more bursty arrival pattern. This explains why the PG unshuffled curve in Figure 6.30 enjoys a larger rate of decrease in its average access delay curve than its shuffled counterpart. The RMP system is not permitted to utilise the DMS feature as fully as the PG systems, by its priority assignment mechanism which, apart from destroying some of the original stream correlation, also lengthens each station's high priority interarrival period for a given load. Therefore, the RMP access delay curve enjoys the lowest rate of decrease with higher load. Note that the RMP system lengthens the arrival period of any priority traffic within an individual station, since it randomly “tags” each arriving message with one of three priorities with equal probability. Hence, on average any given priority will only arrive after two other priority messages have arrived at the station’s queue.

The final observation (iv) is very related to the phenomenon noted in (iii). That is, because of its burstier intra-station arrival profile, the unshuffled PG system will in relative terms derive the most significant benefit from increased usage of the DMS slots, due to the increased probability of *misbehaving queues* (i.e. non-empty for significant periods of time). In addition to this, by observing the top portion of Figure 6.30 we note that the unshuffled PG system was the only one out of the three whose delay rapidly rose to infinity at a load of 1.0; the other two systems did not rise above a delay of 50 slots even at extreme overload (1.55). Hence, it stands to reason that the originally worst-performing system will enjoy the most visible benefit, especially when the property of the system which made it originally the worst (i.e. its intra-station correlation) now (under a cross-correlated arrival process), makes it the best in terms of DMS minislot usage rate, as compared to the other systems.

6.5 Numerical Analysis of Selected Deadlock Models

We begin our analysis in Section 6.5.1 by first considering the impact of various system parameters on the signalling channel performance, for the Basic Deadlock model. After we study the behaviour of this basic model, we explore additional techniques for alleviating deadlock problems, by way of either enhancing or more efficiently utilising the signalling channel capacity. Section 6.5.2 then gives us more insight into possible deadlock scenarios, by providing a numerical analysis of three more comprehensive deadlock models, which model background traffic noise, as well as extreme inter-station correlation. These are the Bernoulli (BER), Machine SerVice (MSV) and Binomial (BIN) Deadlock models.

6.5.1 Performance Evaluation of the Basic Deadlock Model and Signalling Capacity Allocation Schemes

Numerical results are presented here for the Basic Deadlock model derived in Chapter 4, and for the three proposed signalling allocation schemes from that chapter. The aim is to study the effect on F-CPR protocol resiliency, of (i) the parameter p, (ii) the number of CMS's per timeslot for the basic FCS scheme, (iii) the probability of CMS errors, P_{err} and (iv) the number of separate contention resolution groups, k, under the CCS_S scheme. In addition to this, we also compare between the FCS and CCS_M schemes, to determine whether full or partial resource sharing is more efficient, given that the number of CMS minislots is kept identical.

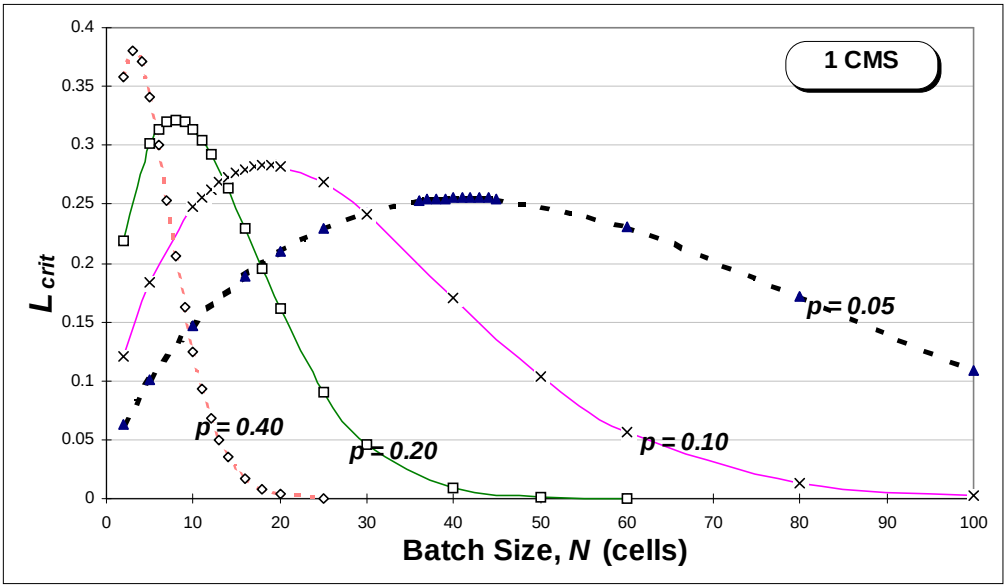


Figure 6.31: Finding an Optimum Batch Size for a given *p*, FCS (*M*=1) Scheme

We begin by observing the behaviour of the critical load as a function of the number of stations for different values of p, without considering the presence of signalling channel errors (P_{err} = 0 in Figure 6.31). We are initially unconcerned with the particular signalling capacity allocation scheme being implemented, so we use the simplest and *default* scheme: the Full CMS sharing scheme, with only one CMS minislot (i.e. FCS with M=1). In the ensuing figures and discussion, note that unless we are focusing on the performance of some specific signalling scheme, this default FCS scheme is assumed.

As shown in Figure 6.31, aggressive p-persistence (large values of the parameter p) allows a greater critical load (L_{crit}) when the batch size (the number of simultaneously contending stations) is small. On the other hand, a low level of p (e.g. $p=0.05$) which provides reasonable protection against deadlock (notice that 50 contending stations transmitting small messages can achieve utilisation of 0.25 without deadlock), achieves quite a low L_{crit} for a small number of contending stations. Nevertheless, the reader is reminded that in such a scenario where a small number of simultaneously contending stations are heavily loaded, a DMS is more likely to be used and higher overall signalling throughput can be achieved, due to the reduced pressure on the contention-based signalling channel.

These results suggest that where it is impossible to vary p in *real-time* HFC system operation, a relatively high level of p should be used, given the presence of the DMS contention-free minislots. The issue of stability is an interesting one in this setting. Recall that if the assumptions of this model were slightly augmented, to apply to an infinite population generating Poisson arrivals, we would have instability for any non-zero arrival rate. However, the arrivals are not Poisson - they are simultaneous batches of N requests. Also, there is no infinite pool of stations - rather, just N stations. Hence, we have a system where the mean CRI length is theoretically finite but can, under certain conditions be so large, that it is for all practical purposes infinite. A practically infinite T_C leads to a near-zero throughput level. From Figure 6.31 we see that this undesirable scenario can happen when the fixed value of p is far from its optimal value (given in Figure 6.33): if p is too low, a small batch will take unnecessarily long to be cleared; if p is too high, a large batch of messages will result in repeated collisions that maintain the backlog at a high level for a long time.

In Figure 6.32 we demonstrate the effect of the number of CMS's per data slot for the Full CMS sharing scheme (once again ignoring the probability of CMS error). The important message of Figure 6.32 is that increasing the number of CMS's does not provide the desired protection against deadlock for the case of a large number of contending stations. The reader is reminded of the cost of increasing the number of CMS's. Given the various protocol overheads for HFC systems specifically, (discussed in Section 5.2.1), increasing the number of CMS's from 1 to 3, would add another 10% of signalling overhead and decrease the "actual user data" throughput capability of an HFC system. Although the exact numbers may be different depending on frame format, a similar outcome would be observed in a WATM system. Indeed, this method will triple the critical load (L_{crit}) for any number of stations and may provide efficient operation and protection against deadlock - but only for relatively small number of contending stations, as seen in Figure 6.32. However, when the number of contending stations is large, and the critical load approaches zero, tripling the critical load is shown not to be beneficial.

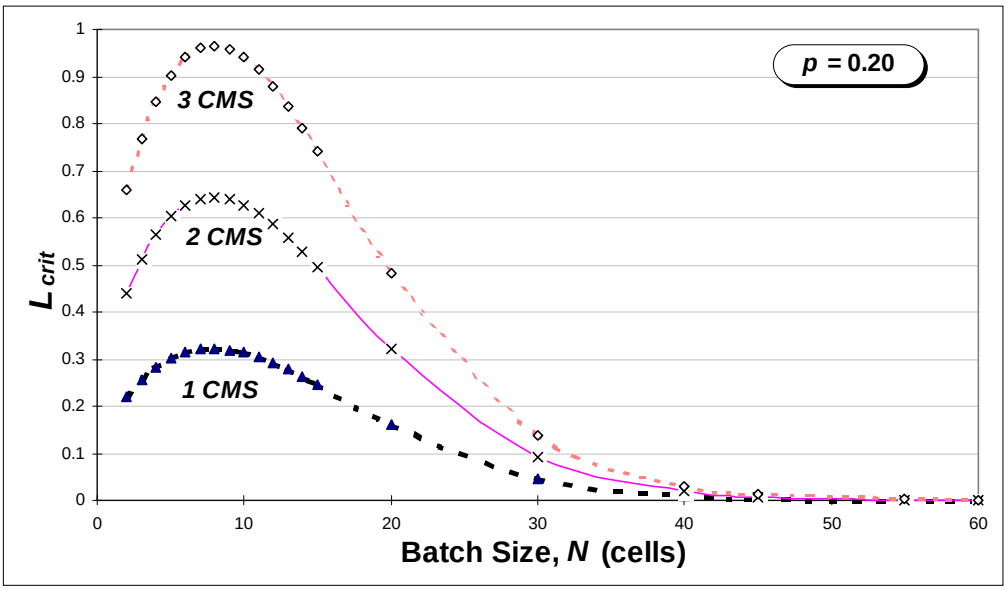


Figure 6.32: Effect of Number of CMS's - FCS (M=1,2 and 3) Scheme

In Figure 6.33 (with P_{err} = 0 once again) we demonstrate that the parameter p not only has a significant effect on the critical load (and on the resiliency of the protocol), but that it can be optimised for maximum load, for each combination of other parameters. A well known fact is that the optimal value of the p-persistence algorithm is 1/n, if the system currently has n contending stations [RIVE 85]. As explained in [SALA 96c], the optimal p for the contention resolution of N initial backlogged requests would be $1/N < p_{opt} < 1$, since, during the contention resolution process the system spends some amount of time in each of the states {N, N-1, N-2 ... 1}, prior to absorption into state zero. This inequality for p_{opt} is clearly illustrated in Figure 6.33. A different way of looking at the inequality is provided in Figure 6.31: for a given fixed value of p, the optimal N is always slightly larger than 1/p (i.e. the previous inequality is reversed, so that $N_{opt} > 1/p$).

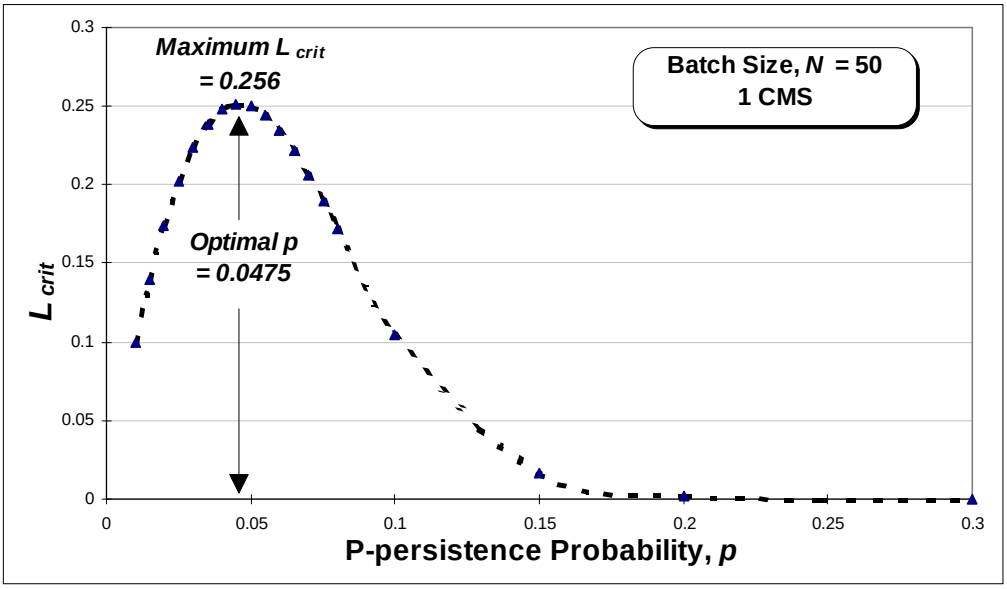


Figure 6.33: Optimising the p-persistence Algorithm - FCS (*M*=1) Scheme

We now turn to look at the effect of the presence of errors on our signalling channel. The graphs in Figure 6.34 highlight the fact that the critical signalling load is largely unaffected by the presence of CMS errors.

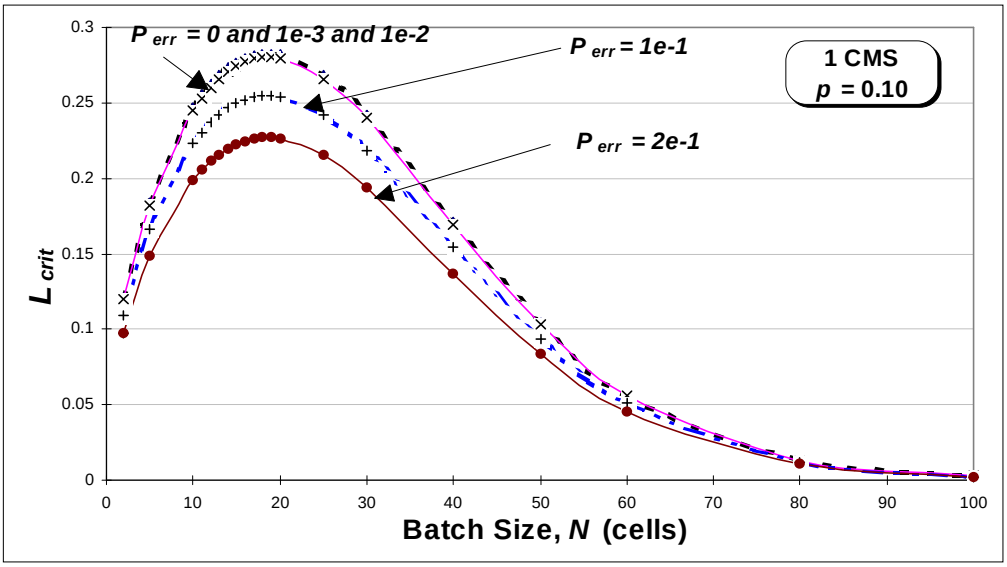


Figure 6.34: CMS Error Sensitivity - FCS (*M*=1) Scheme

At CMS error levels of 10^{-2} or less, the critical load versus batch size behaviour is identical to that of error-free systems, as can be seen from the graph. Note that even an unrealistically high CMS error level of 0.2 causes, at the worst point (N = 20), only a marginal (≈15%) decrease in the achievable signalling throughput level.

Multi-CMS Cyclic CMS sharing, CCS_M, can be thought of as "circuit switching" and it involves more complexity at the station than its FCS counterpart. In particular, with CCS_M, we need the ability for the Head-End to randomly assign the stations to sub-groups; these groups then use only one of the multiple CMS's available. However, as Figure 6.35 shows, the CCS_M scheme yields better performance than FCS since the critical load (L_{crit}) is maintained at a significantly higher level for a larger number of contending stations.

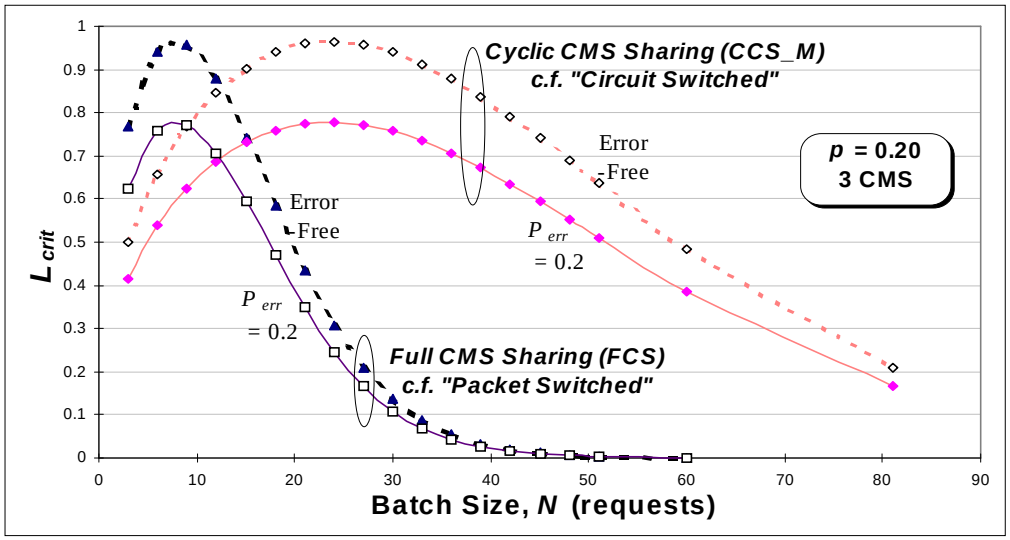


Figure 6.35: Cyclic (CCS_M) vs. Full (FCS) CMS Sharing with 3 CMS minislots, with and without CMS Errors

Note however that as expected, below a certain threshold when the batch size is relatively small (approximately N = 13 in the graph), collisions upon retries are less likely and so it is more efficient to implement Full CMS sharing ("packet switching") and not to waste CMS slots, by cyclically reserving them in TDM-like fashion. As Figure 6.35 illustrates, it is clear that both of these observations apply not only to error-free systems, but also to those with severe (0.2) CMS error rates.

With regards to CMS error-sensitivity, note from Figure 6.35 that even an extreme CMS error level of P_{err}=0.2 causes, for both schemes, a maximum decrease in L_{crit} of only about 18%. However it does appear that the CCS_M scheme is slightly more affected by error over a wider range of N values. In general, the maximum decrease in L_{crit} of 18% is of little impact since a CMS error level of one in every five will be extremely unlikely to ever happen (or be tolerated) in practice, and also since it is a worst-case figure for the range of N values studied. For example, there are other points in the graph where the error-free and errored FCS scheme curves converge to the same line ($N > 35$). A final point to note from Figure 6.35 is that the introduction of errors does not change the shapes of the curves, nor the conclusions drawn earlier about the better overall performance of the Cyclic Multi-CMS Sharing scheme.

In Figure 6.36 we demonstrate the effect of the number of groups k on the critical load for the CCS_S scheme. The synchronous mode of access implicit in this scheme, has similarities with Time Division Multiplexing (TDM) systems. It is therefore quite sensible to also think of k as the *TDM Cycle Length*. Although there is only one CMS per data slot, the access to this CMS is regulated in a TDM-like cyclic manner, so that each station is assigned membership to a group that has a certain position in the cycle (say $TDM_{position}$), and is allowed to access a CMS only at that position and every k (i.e. cycle length) timeslots (notice that k is both the period length and the number of groups). That is, access is allowed at time T, *only if* $(T - TDM_{position}) \% k = 0$. Once again, more complexity would be required within the MAC protocol, both at the stations and Head-End, in order to implement this scheme.

Nevertheless, the CCS_S scheme shows some promise as a method to avoid deadlock, since as Figure 6.36 highlights, for a large batch size (N=60), and a relatively aggressive p-persistence probability, p=0.20, a TDM cycle as short as 6 slots is enough to maximise L_{crit} (except of course for the discontinuity at $k = N$, when $L_{crit} = 1.0$). It has been found that the TDM technique implicit within CCS_S, is good for the alleviation of the signalling congestion created by the extreme inter-station correlation (i.e. large batches of simultaneous arrivals) which we are studying. However, under normal conditions (when the requests are not generated just by single-cell messages, which in turn are not deterministically arriving in simultaneous batches any more, and are more spread in time), one can see that such a TDM technique introduces an amount of unnecessary increase in the average access delay. Hence, a trade-off between cost and benefit exists, the balance of which depends strongly on the traffic profile.

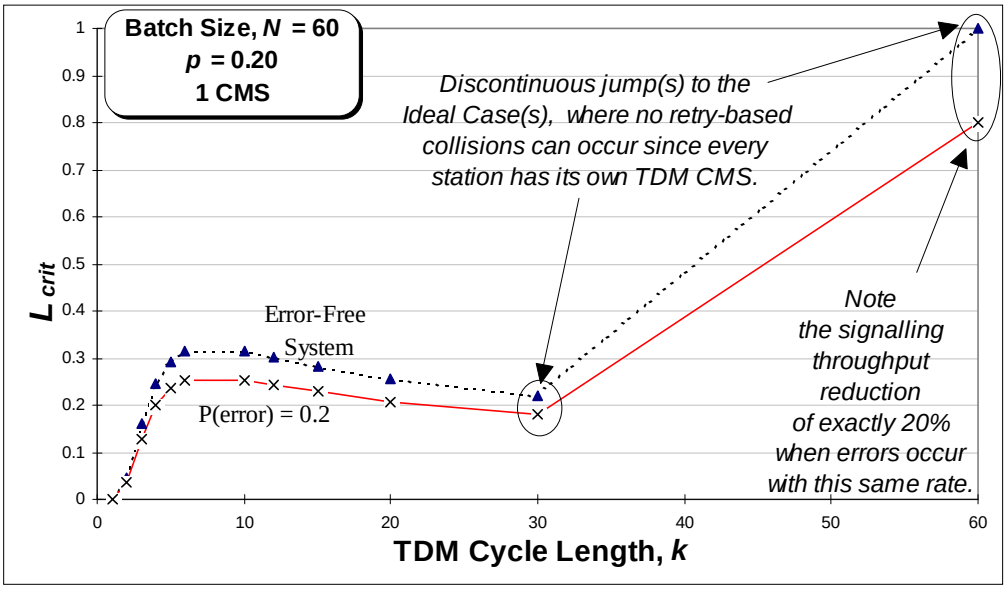


Figure 6.36: Cyclic Single-CMS Sharing (With and Without Errors) - TDM-based access to CMS's

As with the previous two figures exploring the error-sensitivity of signalling performance, Figure 6.36 illustrates the relatively small effect of the presence of errors in the CMS signalling channel, where we once again see no more than an 18% worst-case drop in maximum achievable signalling throughput (L_{crit}), even when the CMS error rate is an unrealistically high 0.2. Also of interest in this graph, is the 20% difference in L_{crit} which may be observed between the very last pair of points, when $k = 60$. While the

dotted curve, representing the error-free case, shows an L_{crit} of exactly one (collision-free operation), the solid curve L_{crit} drops to 0.8, with the error level at 0.2.

We now account for this observation: the CMS minislots are segmented in a TDM-like fashion, and since $N = k = 60$ there can never be any interference between the request arrivals which would cause a collision. The only outcome to require a retry (for any given slot) is an errored CMS, and given that the probability of a CMS error is denoted by P_{err}, we find that the time to successfully clear a single request is geometrically distributed with an average value of $1/(1-P_{err})$ CMS minislots. Therefore, the mean time to clear all the slots is given by $N/(1-P_{err})$, giving a critical load of $N/\{N/(1-P_{err})\} = 1-P_{err} = 0.8$ as seen in Figure 6.36.

6.5.2 Performance Evaluation of the Deadlock Models with Background Traffic

Thus far, we have evaluated the performance of a Basic Deadlock model under traffic conditions leading to signalling channel congestion. That model accounted for the worst-case simultaneous burst arrivals, with and without the impact of errored signalling (CMS) minislots. The study was focused on the impact of the p-persistence and error probability parameters. In addition to our study of the Basic Model, Section 6.5.1 also evaluated different schemes for accessing a varying number of CMS minislots per each upstream data slot. That said, we will not need to study these parameters and schemes again, so in the ensuing results we have assumed a common scenario where only the background traffic model changes, and all else is kept constant. The common parameters used throughout the study of different background traffic models, are given in Table 6.4 below.

CMS Access Scheme	*Full CMS sharing with one CMS per upstream data slot*
Number of CMS Minislots per Upstream Data Slot	*One*
Probability of CMS Minislot being Errored	10^{-3}
p-persistence parameter, *p*	*0.1*

Table 6.4: System Parameters Common to the BER, MSV and BIN Models under Consideration

6.5.2.1 Infinite-source Bernoulli Model - BER

6.5.2.1.1 Average Contention Resolution Interval, T_C

Recall that we have theoretically shown the BER Deadlock model to be unstable for any background traffic intensity λ: the model always yields an infinite T_C. However, since it is the *practical* mean CRI length (as per its definition in Section 4.2.2) which is of interest to us in this study, we shall henceforth be referring only to the practical mean CRI length through use of the term T_C (for the BER Deadlock model).

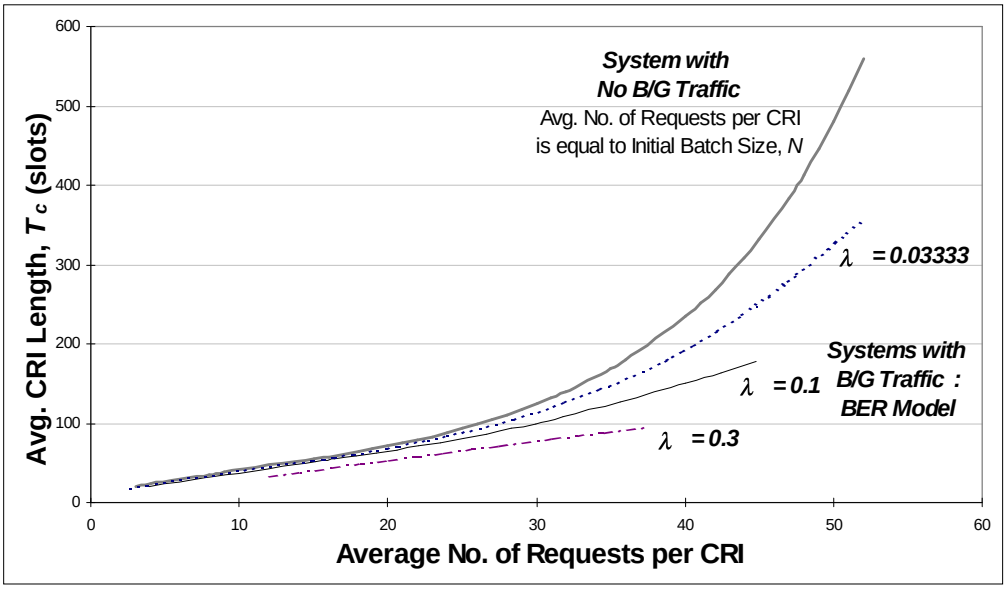


Figure 6.37: BER Model - Impact of Traffic Composition on Average CRI Length T_C , with λ as parameter

Firstly turning our attention to Figure 6.37, we see that the higher the proportion of total traffic made up of "background" requests during a CRI (i.e. the higher the λ), the smaller the T_C, and hence the better the performance of the system. The figure shows that a request load comprising smaller initial batch sizes, N, with some background traffic is resolved faster than its counterpart with the same overall request load, but comprised of a larger N and less newly arriving traffic. Taken to the extreme, the system where the request load is made up only of the initial unresolved batch and no background traffic performs the worst, having the longest T_C under all request load conditions. By next examining Figure 6.38, we make the identical finding that curves for which the initial batch size, N (which in this figure serves as the parameter) is large show a longer T_C.

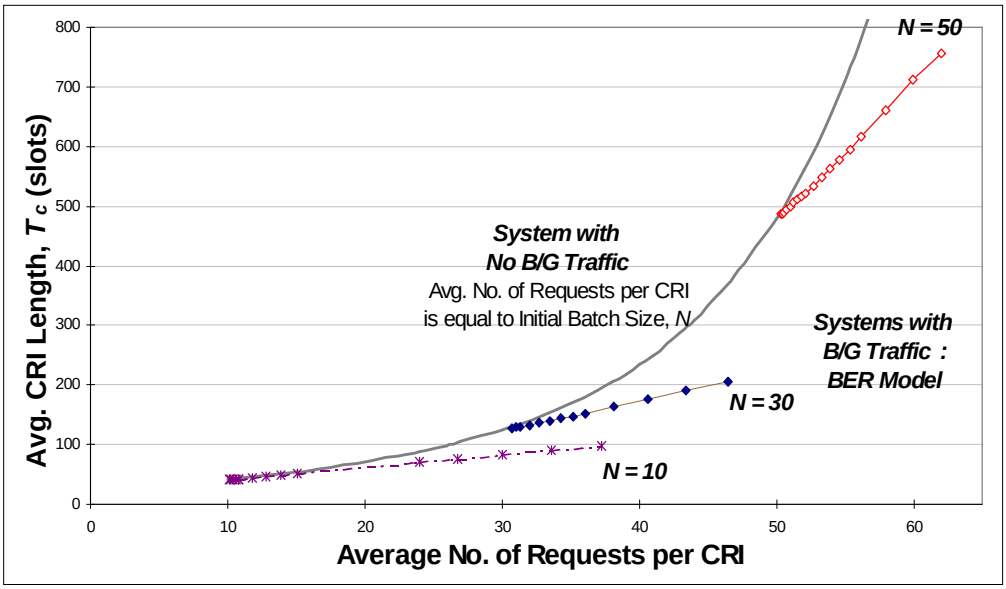


Figure 6.38: BER Model - Impact of Traffic Composition on Average CRI Length T_C , with N as parameter

Once again, the system where the overall request load lacks any newly arriving background traffic and only consists of the initial unresolved batch of requests, performs the worst, having the longest average CRI length. Figure 6.37 and Figure 6.38 emphasise that the average CRI length is dominated by the size of the initial unresolved batch (i.e. T_C depends *exponentially* on N), for a **fixed** p-persistence CRA and system parameters as given in Table 6.4. However, if a scheme which dynamically adjusted p to its optimal value was used, the dependence of the CRI length on N would tend to become *linear* (with gradient e) as $N \rightarrow \infty$ [SALA 96c]. This is significant because use of such an adjusted p-persistence algorithm, together with the background traffic models we propose, would result in the background traffic intensity becoming a dominant factor in the behaviour of average CRI length. At this point, there are no known algorithms for dynamically optimising p in conjunction with the BER, MSV and BIN background traffic models.

6.5.2.1.2 Critical load, L_{crit}

Figure 6.39, where L_{crit} is the ratio between the average number of carried requests during T_C and T_C itself, illustrates the same effect as noted in Figure 6.37 earlier. In this instance, the greater the proportion of the request load made up of newly arriving traffic during T_C, the greater is the maximum achievable signalling throughput, L_{crit}, as a direct result of the shorter CRI lengths discussed earlier.

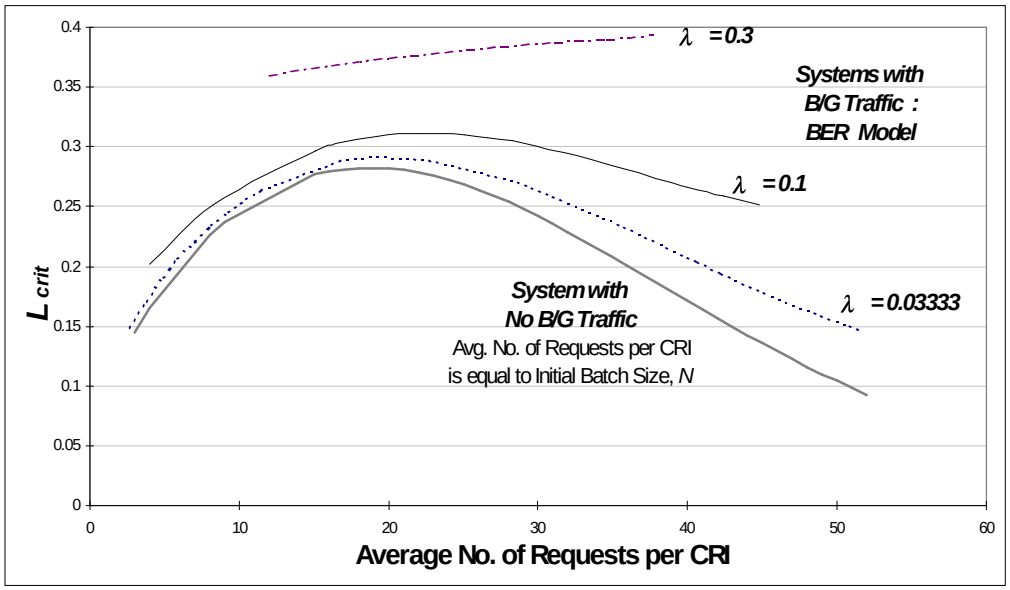


Figure 6.39: BER Model - Impact of Traffic Composition on Critical Load L_{crit} , with λ as parameter

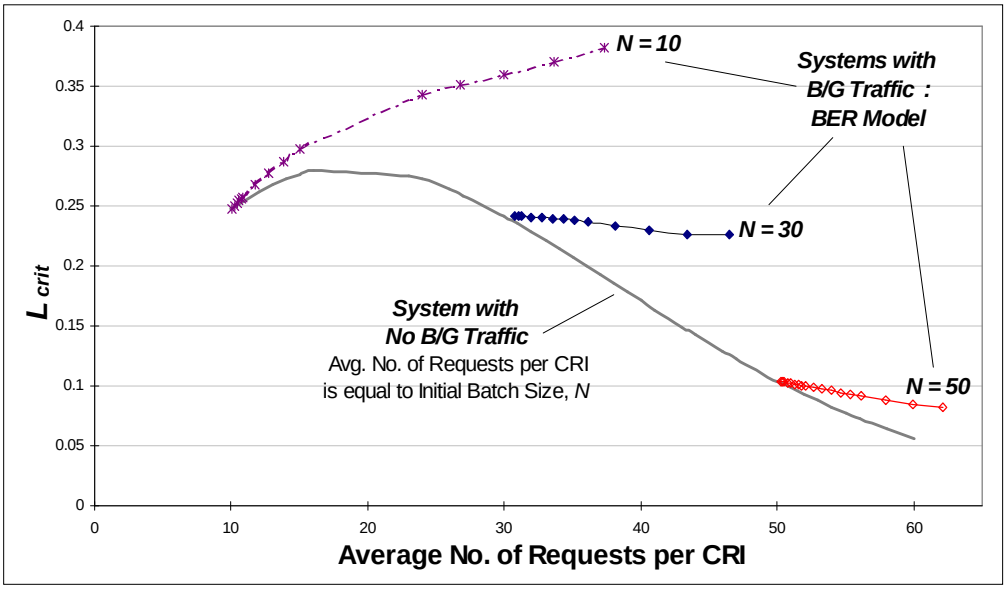


Figure 6.40: BER Model - Impact of Traffic Composition on Critical Load L_{crit} , with N as parameter

A second observation is that the systems where the request load is made up of more background traffic have L_{crit} curves which are more "flat" - thus more insensitive to changes in request load. In particular, once the peak L_{crit} value has been reached, systems with larger λ tend to show a markedly smaller rate of decrease in L_{crit}, as the request load increases. This last point suggests that a system with an overall request load made up of more background traffic and smaller initial unresolved batches, will experience a growth in T_C which is linearly proportional to any growth in the request load per CRI. On the other hand, systems with less background traffic and larger initial batches are more adversely affected, so that growth in T_C occurs at an exponentially increasing rate as the request load per CRI increases. This finding is further reinforced by Figure 6.40, which shows that for sufficiently small initial batch size, say N=10, T_C actually increases slower than linearly with more request load, and thus causes a positive gradient for the L_{crit} vs. load curve (recall that L_{crit} is the ratio of carried request load to T_C).

6.5.2.1.3 Probability of Absorption (Obtaining a finite T_C)

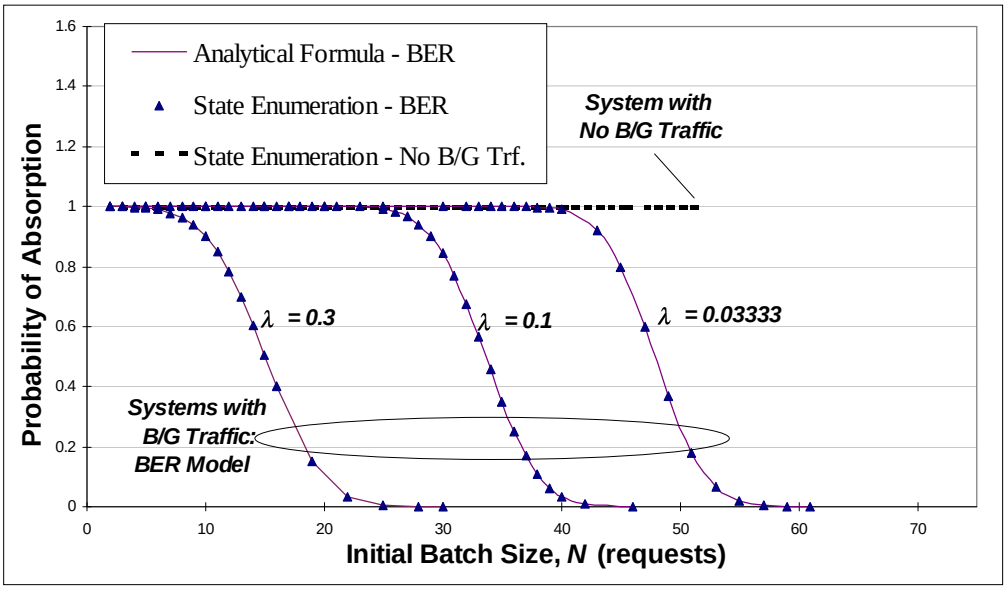


Figure 6.41: BER Model - Impact of Traffic Composition on Absorption Probability, with λ as parameter

The sets of curves within Figure 6.41 and Figure 6.42 illustrate the impact of the Bernoulli background traffic on the system's probability of absorption into state 0, or put another way, of a given CRI instance having a finite length (although the theoretical mean of all possible CRIs, T_C, remains infinite for this BER model).

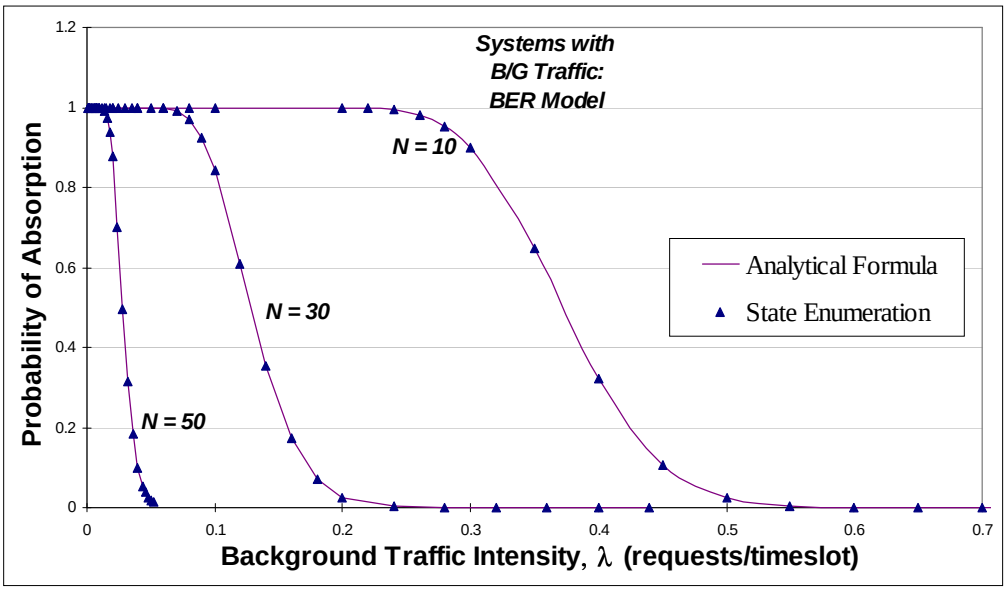


Figure 6.42: BER Model - Impact of Traffic Composition on Absorption Probability, with *N* as parameter

Note that the system without any background traffic always has finite T_C as expected, since the rate of increase of unresolved requests is invariably zero and hence the upper bound of the state space is always finite, and equal to *N*. Figure 6.41 shows that the heavier the background traffic, the smaller is the initial

batch size, *N*, allowable for the system to still have a practically finite mean CRI length (i.e. to remain *practically stable* as per our earlier definition of Section 4.2.2). Looking at it from a different perspective, Figure 6.42 tells us that the larger the initial batch size, the smaller is the allowable background traffic needed to force the system out of the region of practical stability. As an aside, both figures highlight the excellent match between the analytical and numerical methods of obtaining the Probability of Absorption. As expected, the recorded accuracy was to within ε, which was our accuracy threshold.

6.5.2.2 Comparison between an Infinite-Source (BER) and Finite-Source (MSV) Model

The aim of this section is to compare the BER and MSV models, under identical scenarios, in order to highlight the different behaviour of the two models which arises due to the different sizes of their state-spaces. Namely, while the BER model has an infinite state-space, the MSV model may only occupy *L* states other than state 0, where *L* is the total number of stations. Furthermore, by identical scenarios, we mean that (i) only the parameters from Table 6.4 are used for both the MSV and BER models; and, that (ii) the values of the initial batch size, *N*, and the background traffic intensity, λ, are kept the same for both models. In the comparisons which follow, we chose to parameterise the initial batch size, *N*, and vary the background traffic intensity, λ, because this gives a more explicit insight into how different-sized batches behave in a background traffic environment, for the two models. Note that parameterising λ and varying *N* would have yielded results which, although presented from a different perspective, would have been essentially the same.

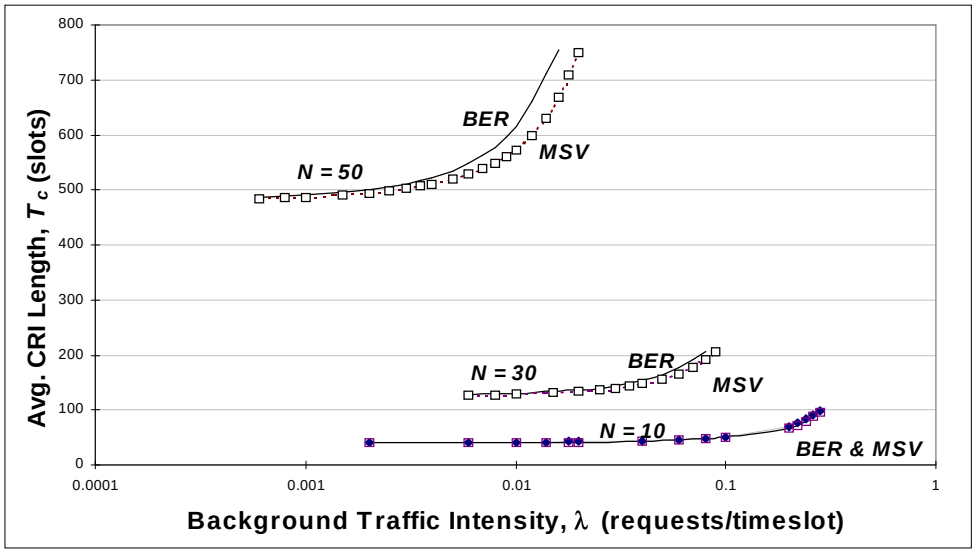


Figure 6.43: Comparison of T_C for Finite-Source (MSV) and Infinite-Source (BER) Models

When comparing the mean time to absorption from initial state *N*, Figure 6.43 illustrates that for small *N* (=10), there is no difference between the two models. Recall that in the case of the theoretically unstable BER Deadlock model, with T_C we are referring to the practical mean CRI length.

The MSV model is stable for any background traffic arrival rate, so in its case T_C refers to the theoretical mean CRI length. The BER model only begins to be discernibly worse at about *N*=50. This reaffirms our earlier finding, that as expected the dominant factor in determining T_C (for the fixed p-persistence

parameter used, p=0.1) is N. In a recursive “cause and effect -type scenario", a large N gives a longer T_C, regardless of the model used. A longer T_C in turn gives the infinite-station, unlimited-state BER model even more time to make its impact and prolong the T_C further than it may be for the MSV model.

Recall from equation (4.23) that the MSV model’s mean arrival rate is always less than or equal to the BER model’s constant, state-independent mean arrival rate λ. Therefore, when the expected time for an initial batch of outstanding requests to be resolved is long, the accumulated discrepancy between the average number of new background requests generated by the MSV and BER model widens, due to the longer time period. This then gives the MSV model the edge in terms of performance, for large initial batch sizes.

Unlike the previous figure, Figure 6.44 shows that interestingly, the biggest observed difference between BER and MSV models occurs with the smallest batch size, N=10. Perhaps counterintuitively, the infinite-state BER model achieves a higher L_{crit} for small N. This is explained by the fact that, as we have just seen in Figure 6.43, for small N, while the T_C for both models is the same over all loads, the mean arrival rate per timeslot is greater for BER. Thus, in about the same time, T_C, the BER model "pushes through" more traffic than the MSV model.

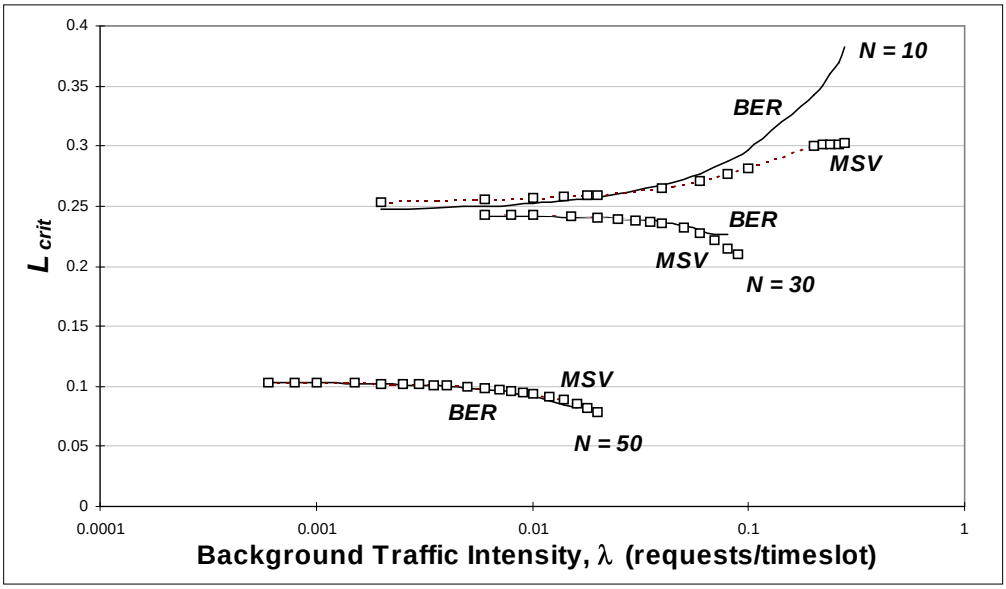


Figure 6.44: Comparison of L_{crit} for Finite-Source (MSV) and Infinite-Source (BER) Models

However, as N increases, so too does the order of magnitude of T_C, and the ever greater adverse impact of BER causes its critical load advantage to be eroded, to the point where at N=50, the MSV model actually has a better L_{crit} for all λ. This arises because L_{crit} is a ratio, and its denominator T_C becomes significantly greater for the BER model than for the MSV model, for large values of N. Not even the extra carried background traffic which the BER model achieves (in the ratio’s numerator), can compensate for the much longer T_C time it gives rise to. The result, as we see in Figure 6.44, is a trend whereby, with larger initial batch size, L_{crit} for the MSV model improves against and ultimately overtakes L_{crit} for the BER model.

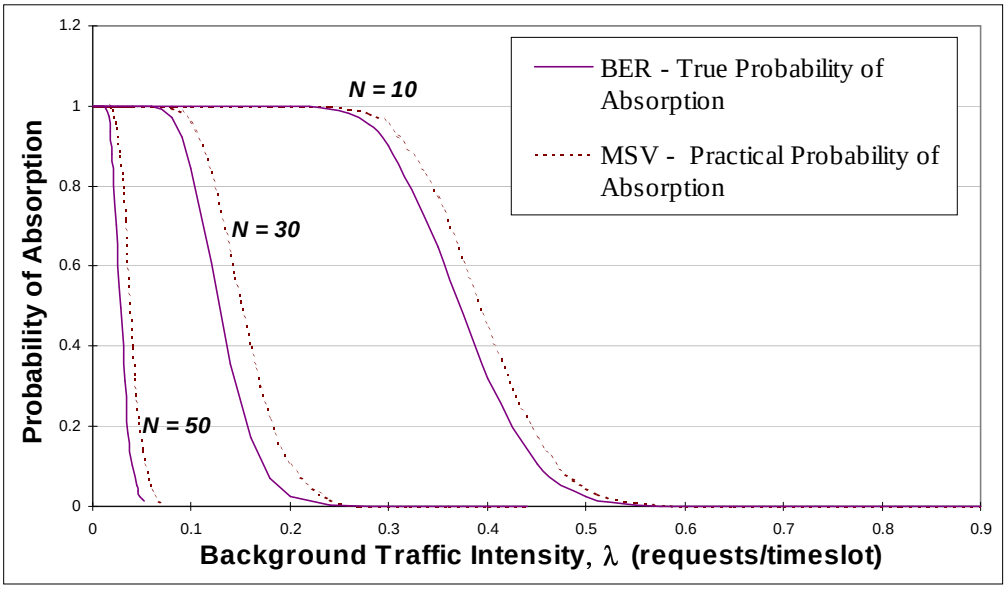


Figure 6.45: Comparison of Absorption Probability for Finite-Source (MSV) and Infinite-Source (BER) Models

It should be stated that two different types of probability of absorption are considered in Figure 6.45. The first is the "True" absorption probability, which may be obtained by the closed-form analytical expression for the BER model (equation (4.17)), and which evaluates the probability that a given instance of a CRI will be of finite duration (recall that the mean CRI is infinite for the BER model). The second is the "Practical" absorption probability, which is calculated in (4.43) for the MSV model, for purposes of comparison with the BER model. Nominally, the true absorption probability for any limited-state model such as the MSV is always 1, regardless of parameters and the initial state. However, as will be shown in Figure 6.50 later, the theoretical T_C is so large for some parameter and initial state combinations, that for all practical purposes it represents infinity. We thus defined in equation (4.43) the new measure called *practical probability of absorption*, in order to calculate the probability that absorption has taken place, by a certain time $t = T_{pr}$ which is practically representative of infinity.

Turning to the curves in Figure 6.45, for all N there is a clear difference between the two models. As one would expect of an infinite-state model, the BER model's curves have a *breaking point* at a smaller value of λ than the MSV model curves, for each of the values of N studied. A breaking point is that value of λ where the absorption probability rapidly starts decreasing from its initial value of one, towards zero. The BER model's earlier observed breaking points are sensible, since the MSV model always generates a smaller mean amount of per-timeslot background traffic (recall equation (4.23)), and also since the MSV model's state space is limited. The latter point in particular tends to improve absorption probability, since there is no possibility of the system state wandering to infinity (i.e. no chance of a system tending towards a limitless number of unresolved collided requests).

6.5.2.3 Comparison between Finite-source models: The Machine Service (MSV) and Binomial (BIN) Models

Having previously studied the effect of state-space size on model performance, this subsection's aim is to investigate how a model's new request arrival process determines its overall behaviour.

6.5.2.3.1 Average Contention Resolution Interval, T_C and Critical load, L_{crit}

Varying N with λ as Parameter

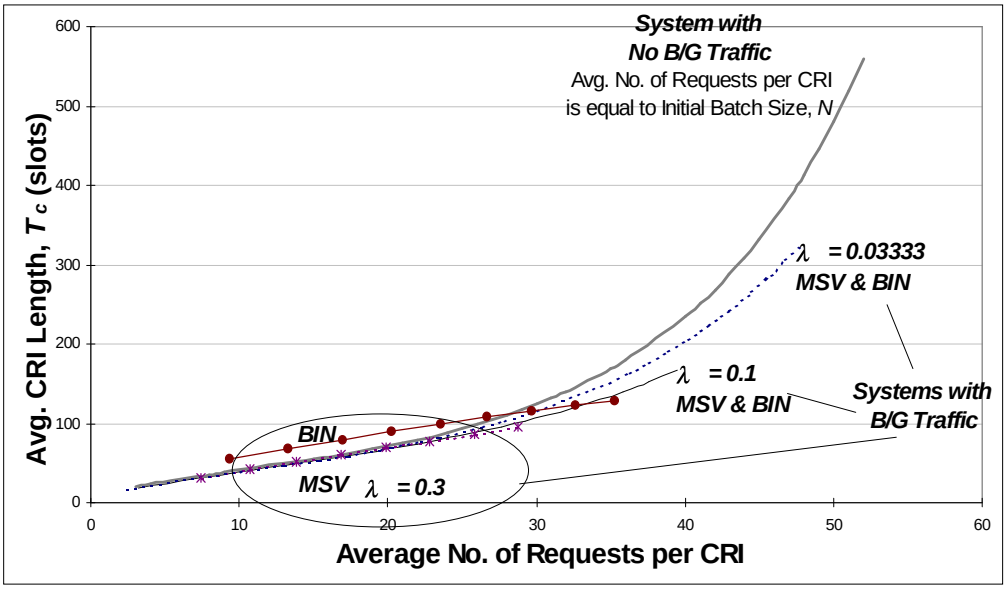


Figure 6.46: MSV and BIN Models - Impact of Traffic Composition on Average CRI Length T_C, with λ as parameter

Figure 6.46 and Figure 6.47 both illustrate that the differences between the BIN and MSV models only come to the fore when the background traffic intensity, λ, becomes sufficiently large (λ=0.3 was sufficient with the other parameters chosen in our study).

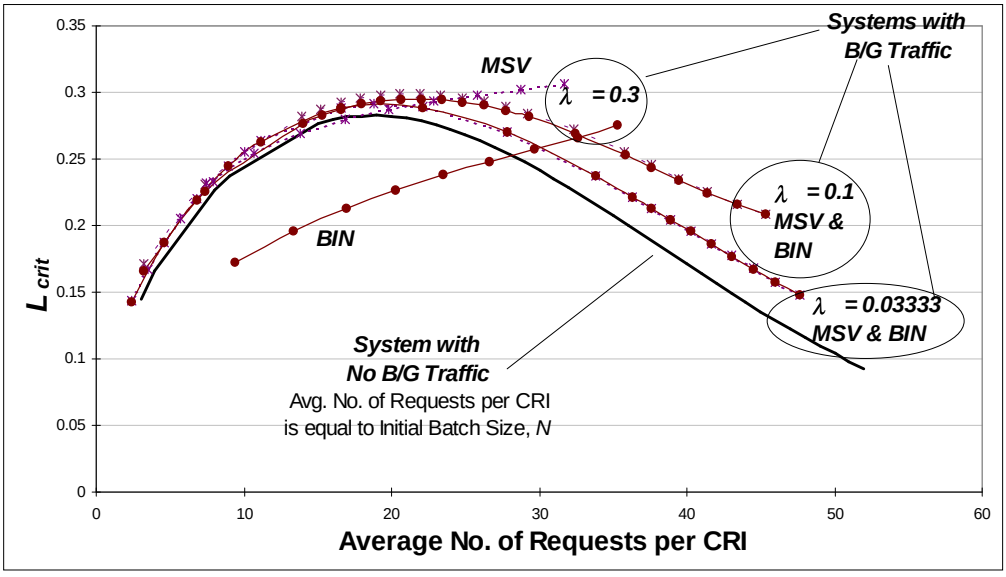


Figure 6.47: MSV and BIN Models - Impact of Traffic Composition on Critical Load L_{crit}, with λ as parameter

However, the most important observation to be made here is how much worse the performance of the BIN model is, at such high background traffic request loads. Inspection of Figure 6.46 and Figure 6.47 tells us that when λ=0.3, the BIN model not only performs worse than the MSV model, but it also has a clearly longer T_C and smaller L_{crit} than the system with no background traffic at all, over a wide range of request loads! This is very interesting, because such behaviour is not observed for either of the other two background traffic models, BER and MSV. As an example for λ=0.3, if we take a sample load of 20 requests per CRI, the BIN model traffic will be approximately made up of an initial batch of N=5 requests, plus another 15 requests which arrive during T_C, which is about 89 timeslots in duration. On the other hand, if N=20 and we have zero background traffic during T_C, the duration of T_C is only about 71 timeslots. In cases like this, it would appear that unlike our earlier findings for the MSV and BER models, the BIN model's background traffic parameter, λ, affects the mean time to absorption and critical load behaviour of the system more strongly than the initial batch size, N.

The reason behind this important observation lies in the BIN model's more aggressive nature of traffic generation: it permits multiple arriving requests in one timeslot. The BER and MSV models describe a system where an initial batch of requests is being "served" and removed from a virtual queue, while intermittent arrivals of at most one new message are recorded during the contention resolution interval. Conceptually, the reason why *the BIN model describes a system with a lower bound on performance*, is that it permits the arrival of new batches even after we have started to serve the initial batch of size N.

As mentioned, when λ is smaller than 0.3, there is no discernible difference between the behaviour of the MSV and BIN models, suggesting that below this combination of background traffic level and p-persistence parameter p, the system behaviour for both models is, as in earlier findings, chiefly determined by the size of the initial unresolved batch of requests, N. Figure 6.46 also illustrates that for λ=0.3 there is a cross-over request load, at which point the T_C curve for the BIN model falls below that of the "no background traffic" system.

Earlier we made the statement that an optimised p reduces the effect of initial batch size as compared to the effect of background traffic intensity. From the presented results, we can infer that, when adjusting the value of p for any of the deadlock models discussed, we are faced with the following compromise:

- Opting for a value which is *too small* will allow the background traffic to dominate, with the BIN model's new batches arriving more often than retry attempts are made. Note that in this situation the MSV and BER models would not have as much to lose, since any new arrival will always be a single message (not a batch!), which, if it finds a free channel, is immediately cleared! Also, the system with no background traffic would not suffer as greatly as the BIN model, since a very small p would only cause the retention of old requests, without any new arrivals.

- On the other hand, choosing a value of p which is *too large* causes degraded performance for all models, due to the incessant "retry-collide cycle" which starts taking place.

This compromise in choosing p highlights the consequences of estimation error that would face any dynamically adjusted p-persistence CRA, which would work by estimating the optimal p on a slot by slot basis.

Varying λ with N as Parameter

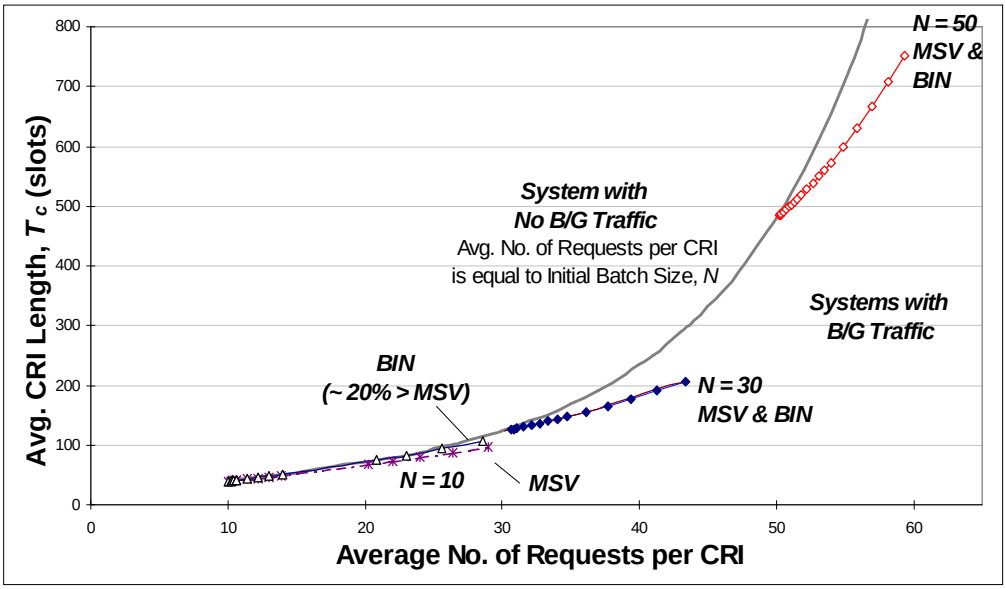


Figure 6.48: MSV and BIN Models - Impact of Traffic Composition on Average CRI Length T_C, with N as parameter

The set of curves in Figure 6.48 and Figure 6.49 reinforces the preceding discussion about the measure of the impact on system performance of the parameters, N and λ.

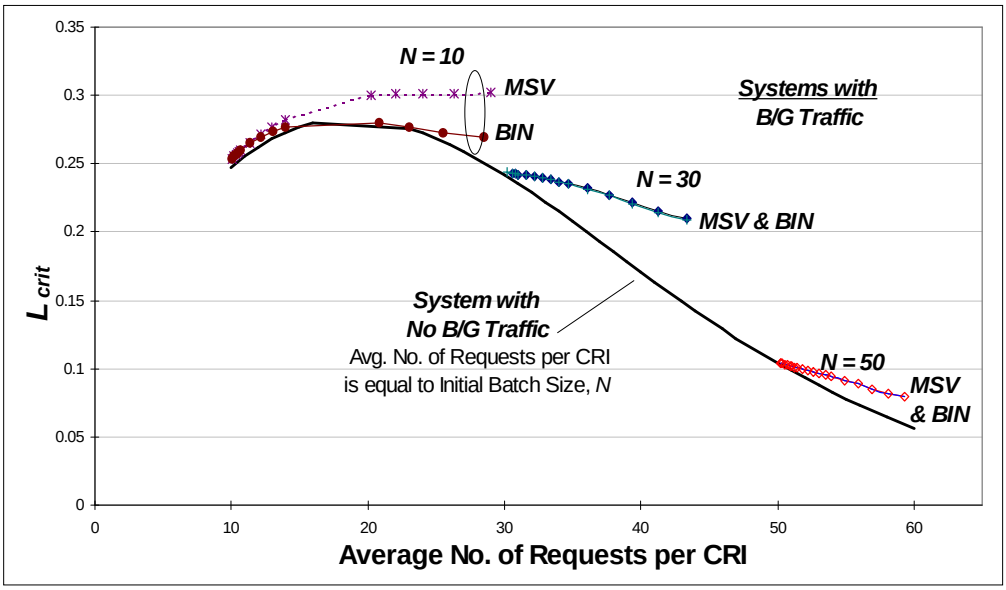


Figure 6.49: MSV and BIN Models - Impact of Traffic Composition on Critical Load L_{crit}, with N as parameter

Namely, for the particular value of p chosen here (0.1), when N is greater than 10 outstanding requests, system behaviour both in terms of average CRI length and the maximum achievable signalling load, is

almost totally dominated by the initial unresolved batch size, and the more aggressive background traffic-generating BIN model has a very minor contribution towards the ultimate value of T_C. As soon as N is sufficiently low for the background traffic process to have a substantial impact on the T_C value, the worse performance of the BIN model as opposed to its MSV counterpart, becomes clearly evident from both Figure 6.48 and Figure 6.49 (see the N=10 curves). It should be pointed out that the BIN and MSV models generate, on average, the same amount of per-timeslot background traffic. Therefore, the total average request load carried will be identical when comparing between the two models; hence, only the duration of T_C affects the critical load ratio, L_{crit}. Interestingly, this is the opposite of what was seen when we were comparing the BER and MSV models. In that instance, it was found that although the BER model generated more background traffic per slot than MSV (on average, with matched parameters), T_C was almost totally unaffected by this fact for small N, so that counterintuitively BER ended up giving a higher L_{crit}. The different characteristics of the three models in question explain this observed phenomenon:

- The comparison of the MSV vs. BER models, is a match-up where the models have the *same traffic generation properties* and thus achieve the same T_C for small N. However, the limited number of stations in MSV reduces that model's average background traffic per slot to a value less than that of the BER model, so the L_{crit} ratio is quite different for small N. As the N increases, BER model's T_C becomes progressively worse than that of the MSV model, thereby reducing the difference in the L_{crit} ratio.

- The comparison of the MSV vs. BIN models, is a match-up where although the models have *the same average background traffic per slot*, they have different traffic generation characteristics. Given that the effect of background traffic is only visible for small N, it is not surprising that this comparison only shows differences in T_C and L_{crit}, when N is small.

6.5.2.3.2 Mean Step Duration and Measure of "Upwards Pull" in the Upper Regions of the State Space

Before we present results for the practical probability of absorption comparison between the MSV and BIN models, it is important to examine two facets of the models' behaviour which directly influence the PPA outcomes:

- The first of these is the mean descending step duration, for a system which is in the upper regions of the state space. We are interested, for the parameters in this study, in how long on average it takes to reach state j-1 from state j for the first time, when j is very close to the total population limit L. This is a very significant indicator of how long a system is going to be "trapped" in the vicinity of L (which we will soon see to be a *quasi-absorbing* state itself), and thus for what period of time it will be unable to return to state 0 and be absorbed.

- The second factor affecting the *PPA* curves is complimentary to the first, in that it looks at how a system behaves in its initial stage of evolution, having started in state $j=N$ at time t=1. By observing

this *transient* behaviour, we gain insight into how probable it is for the system to initially climb to and subsequently be trapped in (for a long time), the vicinity of *L*.

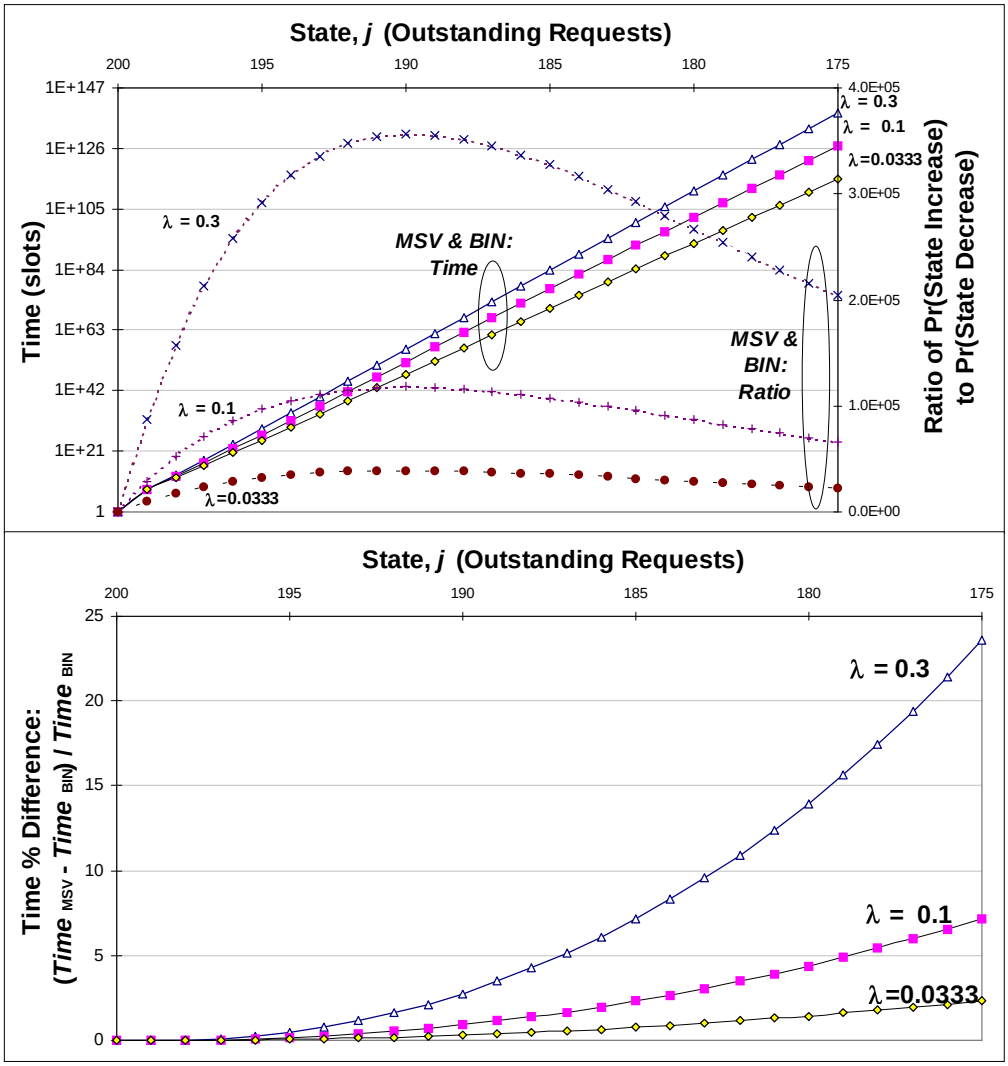


Figure 6.50: Mean Step duration and "Upwards Pull" in Upper Regions of the State-Space

Let us start by focusing on the first of these factors, the essence of which is captured in Figure 6.50. Note that in this figure, we have presented not only the mean step duration and percentage difference in mean step duration (between MSV and BIN), but also a measure of the upwards tendency of the system, when in state *j*. This latter measure, called the *upwards pull*, is a ratio between the probability of state increase to that of state decrease. Of course, for the BIN model, the probability of state increase is made up of many separate increase events (recall that in state *j*, system occupancy can increase by 1 or 2 or ... *L-j* outstanding requests).

The most prominent part of Figure 6.50, depicted in the upper portion, is the sheer magnitude of the average waiting times to descend down from the upper-bound state *L*, to some of its closest neighbours, let alone to a state "only" 25 transitions away, such as *j*=175. The time to reach the latter states exceeds 10^{100} timeslots, making the *j*=*L* state an absorbing one for all practical purposes and hence allowing us to name it "quasi"-absorbing. The magnitude of mean descent time is largely independent of the background traffic intensity, reaffirming our earlier finding that for large *N* and the relatively large p-persistence parameter

used (=0.1), the mean absorption time into state 0 is largely independent of the presence of background traffic. Rather, the mean descent time is strongly governed by the infinitesimal likelihood of successful retries using the fixed p-persistence algorithm with so many competing stations and such a large *p*. This circumstance is a prime example of the need to monitor the number of participants in the collision resolution process, and adjust the p-persistence parameter accordingly. As explained in Section 4, while estimation-based techniques for the optimal dynamic adjustment of *p* do exist, they are only applicable under certain assumptions about the traffic (e.g. Poissonian arrivals); these assumptions do not hold for the BER, MSV and BIN Deadlock models we have proposed, and so a dynamic *p* adjustment algorithm for these models remains for further study.

Another interesting feature of the upper part of Figure 6.50 is the non-linear function describing the relationship between the upwards pull and the current state, *j*. As expected, this ratio, representing the state-dependent "upwards tendency" of a system, increases with increasing *j* due to the decreasing likelihood of a successful retry. Opposing this effect, we recall that as *j* increases, the probability of an arrival decreases due to a smaller population of potential generators. The retry probability tends to be the dominant of the two effects mentioned, until a particular point in close proximity of the highest state *L*, where the upwards pull hits a peak from which it drops rapidly to zero at *j=L* (since there are zero potential generators when all *L* stations are already contending). Looking at the top portion of Figure 6.50, the order of magnitude of the mean step times makes it easy to neglect any differences that may exist between the MSV and BIN models.

However, focusing our attention to the lower part of Figure 6.50, we come to see that the mean step time for the two models is actually different; this difference increases in smaller states, and in particular it increases with a larger background traffic intensity λ. In order to better understand this phenomenon, we must examine the upwards pull difference between the two models, over the entire state-space. This is presented in Figure 6.51.

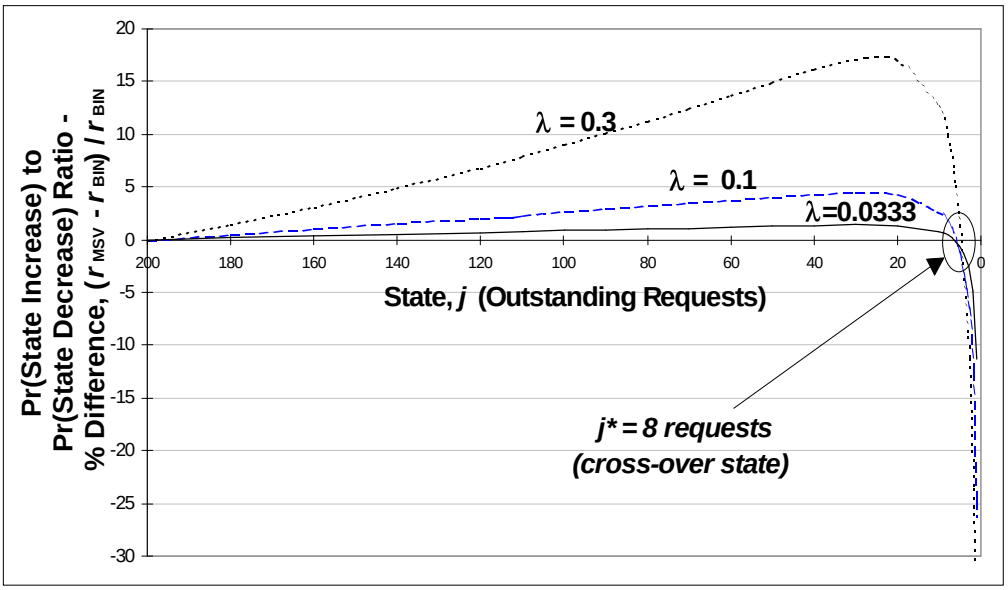


Figure 6.51: Ratio of State Increase/Decrease Probabilities - % Difference between MSV and BIN Across State Space

Figure 6.51 shows that the upwards pull becomes higher for the MSV than the BIN model for $j > j^*$ (where j^* is about 8 outstanding requests, for the parameter combinations we have considered). This change of sign in the difference between the two models arises because the probabilities of a state increase and a state decrease are strongly affected by the differing nature of the models. The key contributing factor to the observed non-linearity and change of sign, is that more than one arrival is possible in the BIN model. This means that for the BIN Deadlock model, the ratio of state increase to decrease probabilities must take into account the sum of all the possible state increase probabilities out of a given state, j. Equations (4.53) and (4.57) are such that for small j, when the Binomial arrival distribution has a very large population $L-j$, there are many potential ways of exiting the state. Hence from $j = 0$ until the "crossover state" j^*, the BIN model's probability sum of state increases is larger than the corresponding single probability of the MSV model's state increasing by one.

In summary, for both the MSV and BIN models, equations (4.49) and (4.83) respectively have shown that the mean step duration is entirely dependent on the upwards pull ratio. This fact is clearly verified by the strong correlation between the mean step duration difference in Figure 6.50 and the upwards pull difference in Figure 6.51, which we have just investigated.

6.5.2.3.3 Transient Analysis

Focusing firstly on the lines in Figure 6.52 (overleaf), we find that across the full range of initial batch sizes studied for λ=0.3, the system converges very quickly (i.e. between 10,000 and 100,000 timeslots) to either:

(i) The true absorbing state j=0, or,

(ii) The quasi-absorbing state j=L.

Note that the composite probability that by time T, the system has either *already been absorbed into state j=0*, or, is *currently in state j=L*, is extremely close to unity, suggesting an infinitesimally small chance of any other outcome at time T. The almost negligible amount of difference between these composite probabilities, for values of T orders of magnitude apart (10^4, 10^5 and 10^6 timeslots) is testimony to the great speed of the system's convergence to one of these states. Having made the above observation about the speed of this convergence, the bar portion of Figure 6.52 then gives further insight into the relative probabilities of actually occupying one or the other of the j=0 or j=L states, at time T.

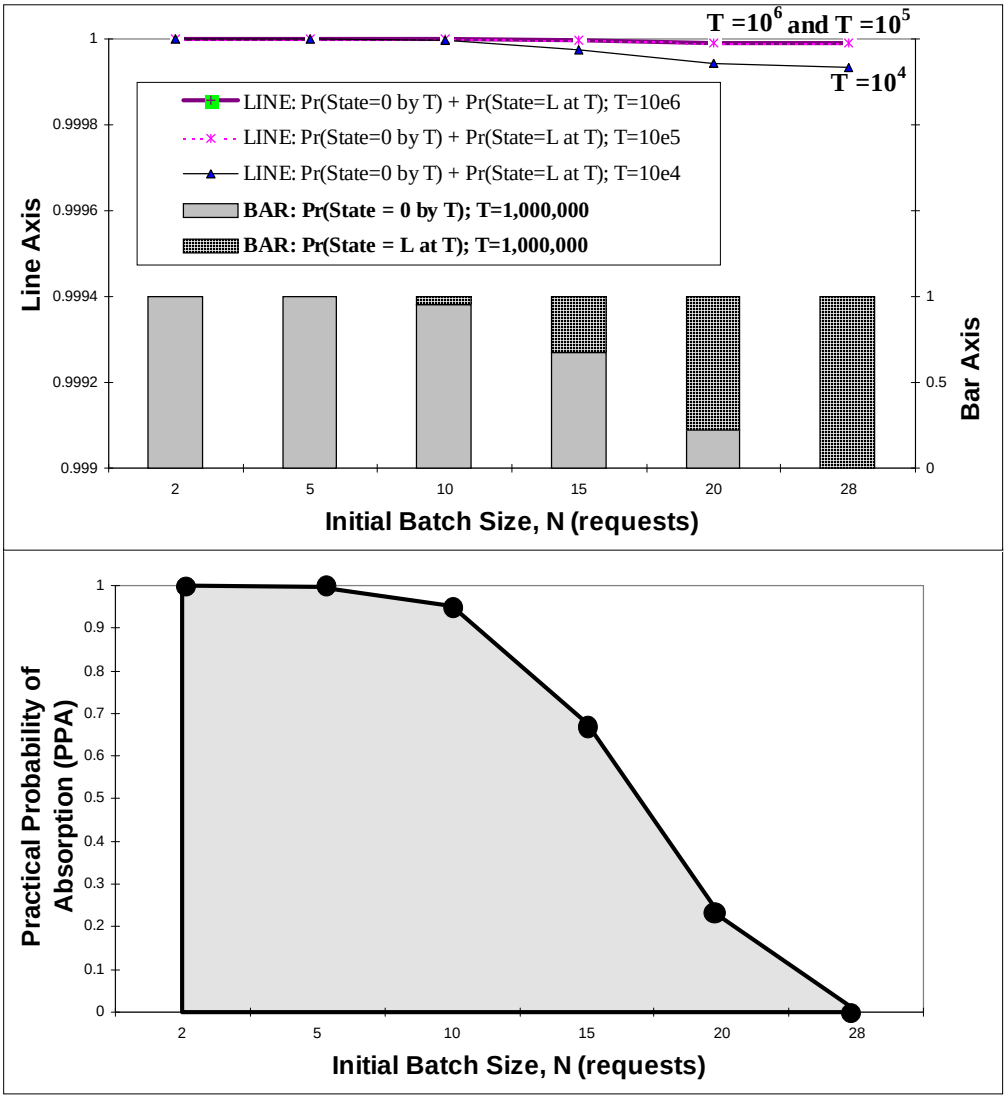


Figure 6.52: Transient Analysis over a Range of Initial Batch Sizes, *N*, for λ=0.3

We have only showed this "occupancy split" for T=10^6 timeslots, since the other two occupancy comparisons yield almost identical results. Two things are of significance when one views the composite-shaded bar lines: (a) as expected, the occupancy split is very sensitive to increasing *N*, with the probability of the system having been absorbed by T at almost 0 for *N*=28 initial outstanding requests; and (b) the practical absorption probability does not vary significantly regardless of whether it is calculated for $t=T_{pr}$ or for *t*=T, as shown in the lower portion of Figure 6.52.

As an example, the bar graphs showed T as 1 million timeslots, and gave exactly the same probability of absorption values as the *PPA* curve below, which was calculated over the standard period of T_{pr} =12.5 million timeslots. This illustrates that the transient behaviour of the system has a very significant impact for what we consider as the "final state distribution", when *t*=12.5 million timeslots, practically speaking, approximates $t = \infty$, as explained earlier. In summary, it has been shown that during its transient stages, for small *t*, the system quickly enters the quasi-absorbing state *j*=*L*, with increasing probability for larger values of initial batch size *N*. A direct result of this transient behaviour is the calculated *PPA* value - in proportion to increasing *N*, we obtain a decreasing PPA, since we are more likely to enter the quasi-absorbing state *L*, from which "escape" is practically impossible. Recall from Figure 6.50 that the mean

time to descend to a state as close as L-25 from state L, is in the order of 10^{100} timeslots (descending all the way to state 0 would take much longer). It becomes clear that there exists an inversely proportional relationship between (a) the probability with which the uppermost state $j=L$ is reached in the early transient stages of system evolution, and (b) the PPA.

6.5.2.3.4 Practical Probability of Absorption

Wholly consistent with our earlier observations, Figure 6.53 and Figure 6.54 (overleaf) show that the most significant difference between the BIN and MSV models occurs when the background traffic intensity is at its greatest relative to the value of N. Therefore, for λ=0.3 and small N in Figure 6.53 and for N=10 and large λ in Figure 6.54, the less aggressive MSV arrival process yields a slightly better PPA curve than the BIN model. That is, it is more likely to be absorbed into the j=0 state within a practically feasible time. These results reinforce our earlier findings (Figure 6.51 in particular), showing that for large background traffic intensity relative to N, the BIN model is more likely than the MSV model to drive the system state higher towards L, and hence less likely to return to the absorbing state j=0.

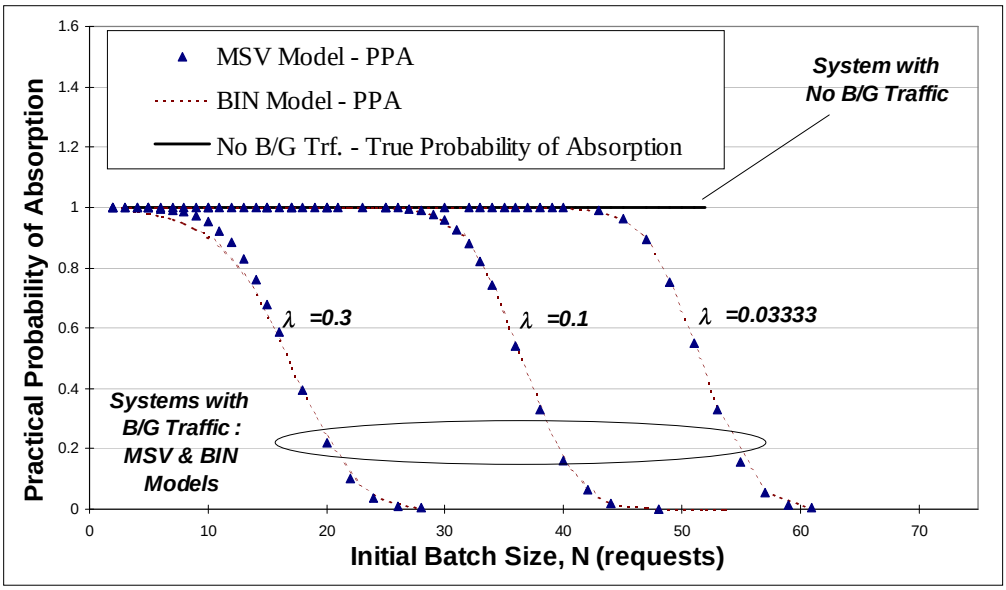


Figure 6.53: Comparison of Practical Probability of Absorption for MSV and BIN Models, with λ as parameter

However, if we closely focus on the tails of all three curves in Figure 6.53 and Figure 6.54, we find that when N is medium to large relative to λ, exactly the opposite kind of behaviour to that just described, may be observed! Namely, in the tail regions of the graphs, the MSV model has very slightly worse PPA values than the BIN model and is therefore less likely to have a practically finite T_C. Figure 6.51 justifies this behaviour in terms of the *Pr(State Increase)* / *Pr(State Decrease)* ratio, which we earlier termed as upwards pull. That is, Figure 6.51 has shown that in regions where N is medium to large as compared to λ, the MSV model's upwards pull is stronger than that of the BIN model, and this difference is even more exaggerated with a larger λ. In addition to this, inspection of Figure 6.50 illustrates that the mean times to step down from the states close to L are up to 25% longer for the MSV model than for the BIN model.

This means that the MSV model not only has a slightly greater propensity to move towards the upper regions of the state space for $j > j^*$, but that when it does so, it is "trapped" in those regions for slightly longer periods of time before sojourning. This accounts for the MSV model exhibiting slightly worse PPA values than the BIN model in the tail regions of Figure 6.53 and Figure 6.54, as observed.

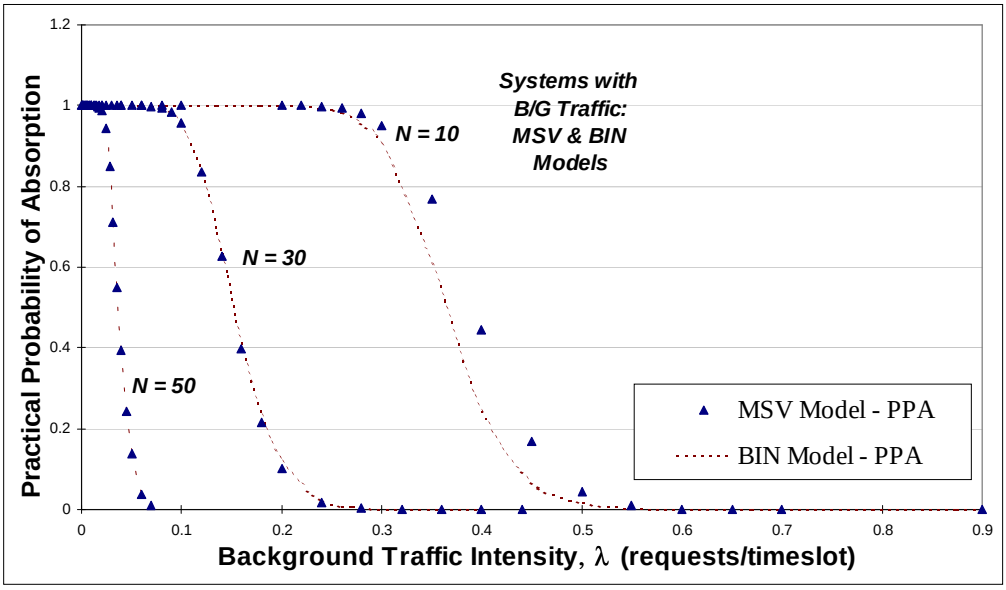


Figure 6.54: Comparison of Practical Probability of Absorption for MSV and BIN Models, with *N* as parameter

7. Conclusions

In this thesis we have investigated in detail a generic multi-service access protocol, called Fair Centralised Priority Reservation (F-CPR) by providing an extensive teletraffic study of its performance and behaviour under realistic traffic conditions; the study has included a set of new models exploring signalling-based protocol deadlock, and a new multi-priority scheduling scheme for the Head-End controller. Through the teletraffic study, we have provided deep insight into the protocol's characteristics and peculiarities under certain conditions, and have synthesised significant extensions which can be either part of, or in addition to, the basic medium access control (MAC) protocol.

Firstly, the thesis has reported a detailed simulation study of the F-CPR MAC protocol, under realistic traffic conditions based on Ethernet LAN traces, with considerable attention being paid to intra- and inter-station correlation. F-CPR has been found to exhibit max-min throughput fairness when loaded by traffic of both a Poissonian and self-similar nature; positional fairness has also been determined, with no physical location within the HFC access network being able to provide a station with more than its fair share of medium access. Perhaps counterintuitively, self-similar traffic with long-range dependence, which would intuitively be expected to adversely affect Ideal Multiplexer performance in comparison to a Poisson traffic load, has been found to actually improve overall system utilisation to a certain degree. This interesting phenomenon has been shown to occur because the highly correlated nature of the real trace causes a greater system-wide probability of at least one non-empty queue, thus enabling F-CPR's contention-free "piggybacked" bandwidth reservation feature, which relies on non-empty queues, to be used more often.

As part of the detailed F-CPR simulation study, we have also demonstrated that excluding disaster scenarios, and extreme inter-station correlation, the F-CPR performs very close to its Ideal Multiplexer benchmark, and hence can be modelled as an Ideal Multiplexer for a wide range of traffic parameters, with increasing accuracy as the number of stations becomes higher. We have shown that the essential difference between an Ideal Multiplexer and F-CPR is that the state knowledge about the individual stations' queues is unknown to the protocol's Head-End controller, while it is always available to the Ideal Multiplexer. This fact has meant that the F-CPR protocol needs the overhead of signalling, and is prone to propagation and Head-End controller processing delays. We have shown that the need for the stations of an HFC system to continually have to signal bandwidth demands, and to have to stop and wait between allocated capacity bursts, makes the F-CPR protocol perform much worse subject to inter-station correlation than the Ideal Multiplexer, which by definition centralises all queue state knowledge.

Our simulation study has highlighted that a large number of active stations with very small messages and light individual station load may lead to signalling channel congestion collapse due to an avalanche of request collisions, particularly when the inter-station correlation is high. Although the presence of F-CPR's contention-free "piggybacked" bandwidth reservation feature has been shown to postpone and sometimes help to avoid signalling channel deadlock, it has been demonstrated that under some conditions MAC protocol deadlock is ultimately reached.

The overall distribution among individual stations of the generated traffic load has also been found to interact with the F-CPR's stop and wait nature between messages, and significantly impact global system utilisation performance. Thus if only a small fraction of the active stations generated a large proportion of the traffic load, overall system utilisation was found to be significantly lower than if the load was more evenly distributed among the stations.

Another significant component of this thesis has been the development of a suite of disaster scenario (deadlock) models and their analysis by means of a discrete-time Markov chain technique, along with the introduction of a concept of contrasting practical and theoretical system (access delay) stability. Our models have been useful in providing a detailed set of conditions which were shown to lead to practical instability and deadlock, and which depend on such factors as signalling channel error probability, the profile of signalling traffic, and properties of the contention resolution algorithm being applied to the signalling channel. We have proposed and tested three new signalling channel capacity allocation schemes, with a view to extending the usable region of the F-CPR protocol, by avoiding deadlock under as wide a range of conditions as possible. We have identified the best-performing of these three schemes, in terms of extending the protocol's practically usable load region the furthest, to be the Cyclic Contention Mini-Slot sharing technique employing multiple CMS's per data slot. We have demonstrated that the Time Division Multiplexing (TDM) -like property of this scheme is the key to its success, especially under the disaster scenario conditions we have simulated through our deadlock models. However, this scheme has also been shown to require the most additional Head-End and station intelligence, as well as some extra signalling bandwidth, to implement.

As part of the third major contribution of this thesis, we have developed a new multi-priority scheduling scheme applied at the Head End central controller, based on a pre-emptive queueing principle. In addition, we have proposed a mechanism for the generation of multi-priority traffic from an (unprioritised) existing trace, based on the principle of having as many station subsets, with different sized populations, as there are priority levels in the system. We have also compared and contrasted the performance of our new multi-priority scheduling scheme, and prioritised traffic generation mechanism, to an existing scheduling scheme and a random-hash based generation mechanism. Significantly, our new scheduling scheme has been shown to perform better in minimising average access delay of high priority traffic under realistic intra- and inter-station correlation conditions.

In the multi-priority scheduling scheme analysis, an interesting counterintuitive finding for both our scheme and the existing scheme, has been that under extreme *inter*-station correlation, the average access delay of the high priority messages may significantly decrease with load. The reason was demonstrated to be that at low loads when the effect of queueing at the Head-End scheduler is negligible, all the priorities are treated equally as they simultaneously "hit" the contention-based signalling channel, which does not discriminate based on priority. Then, as the load is increased the high priority traffic is shielded by its foremost priority status from the negative effect of scheduling-related queueing at the Head-End, while simultaneously enjoying the benefit of an increased usage rate of "piggybacked" bandwidth reservations. In this way, as the high priority queues begin to fill with higher system-wide load, the collision factor is

slowly removed and the importance of priority labels begins to again emerge. It was found that the increased rate of signalling collisions in larger systems made the effect more pronounced than in smaller systems. Deeper probing of this interesting phenomenon has also shown that the degree to which the high priority access delay decreases, depends on the level of *intra*-station correlation of the high priority stations' streams, since high intra-station correlation has been demonstrated to enable F-CPR's "piggybacked" bandwidth reservations to be used at an increased rate.

8. References and Bibliography

[ABRA 70] N. Abramson, "The Aloha system - another alternative for computer communications", *Proceedings of 1970 Fall Joint Comp. Conference*, vol. 37, AFIPS Press, Arlington, VA, pp.281-285, 1970.

[ADDI 95] R. G. Addie, M. Zukerman and T. Neame, "Fractal traffic: measurements, modelling and performance evaluation", *Proceedings of INFOCOM '95*, Boston, April 1995.

[ADDI 96] R. G. Addie, "Traffic will be more Gaussian in future.", *Proceedings of Australian Telecommunication Networks and Applications Conference ATNAC '96*, Melbourne, December, 1996.

[ADDI 97] R. G. Addie, "Multiplexing Gain in Large Networks of the Future", *Proceedings of the Long-Range Dependence International Workshop*, Queensland University of Technology, Brisbane, Australia, 28-30 January, 1997.

[ADSF 97] ADSL Manufacturer List, *The ADSL Forum Home page at http://www.adsl.com.*

[ANSI 95] *Asymmetric Digital Subscriber Line Metallic Interface*, ANSI Standard T1.413-1995

[ARM 90] G. J. Armitage, and K.M. Adams. "Architecture of a Multimedia Desktop Workstation.", *Proc. Australian Video Communications Workshop*, Melbourne, pp. 76-85. July, 1990.

[ARM 93] G. J. Armitage, and K.M. Adams. "Using the Common LAN to Introduce ATM Connectivity.", *Proc. IEEE Computer Society 18th Conference on Local Computer Networks*, Minneapolis, MN. September 19-22, 1993.

[ATF1 96] ATM Forum AF-PHY-0064.000, “E1 Physical Interface Specification”, September, 1996.

[ATF2 95] ATM Forum AF-TM 95-0177, “Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey”, August, 1995.

[BERA 94] J. Beran, R. Sherman, M. S. Taqqu and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", accepted for publication (subject to revision) in *IEEE Transactions on Communications*, 1994.

[BERT 92] B. Bertsekas and R. Gallager, *Data Networks, 2nd Edition*, Prentice-Hall, 1992.

[BISD 96a] C. Bisdikian, B. McNeil, R. Norman and R. Zeiss, "MLAP: A MAC Level Access Protocol for the HFC 802.14 Network", *IEEE Communications Magazine*, vol. 34, no. 3, March 1996.

[BISD 96b] C. Bisdikian, K. Maruyama, D. Seidman and D. N. Serpanos, “Cable Access Beyond the Hype: On Residential Broadband Data Services over HFC Networks” *IEEE Communications Magazine*, vol. 34, no. 11, pp. 128-135, November 1996.

[BISD 96c] C. Bisdikian, “A Review of Random Access Algorithms”, *IEEE 802.14 Working Group Document No. 802.14-96/019*, January, 1996.

[BOND 87] D. J. Bond, "A Theoretical Study of Burst Noise", *British Telecom Technology Journal*, vol. 5, no. 4, pp. 51-60, October 1987.

[BOX 76] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, 2nd ed. San Francisco, CA: Holden Day, 1976.

[CAPE 79] J. I. Capetanakis, “Tree Algorithm for Packet Broadcasting Channel”, *IEEE Trans. Inform. Theory*, vol. IT.25, pp. 505-515, September 1979.

[CHEN 95a] Y. Chen, Z. Deng and C. L. Williamson, "A model for self-similar Ethernet LAN traffic: Design, implementation and performance implications", *Preprint, Technical Report*, Dept. of Computer Science, University of Saskatchewan, 1995.

[CHEN 95b] K-C. Chen and D-C. Twu, “A Novel Medium Access Control Protocol for Broadband Communication over CATV Based Community/Metropolitan Area Networks”, *IEEE 802.14 Working Group Document No. 802.14-95/67*, 1995.

[CIF 96] "Cells In Frames Version 1.0: Specification, Analysis, and Discussion", *CIF Alliance Specification*, edited by Scott W. Brim, Cornell University, October, 1996. *URL: http://cif.cornell.edu/specs/v1.0/CIF-baseline.html*.

[COX 84] D. R. Cox, "Long-range Dependence: A Review", in Statistics: An Appraisal, H. A. David and H. T. David eds., The Iowa State University Press, pp. 55-74, Ames, Iowa, 1984.

[COX 92] D. C. Cox, “Wireless Network Access for Personal Communications”, *IEEE Communications Magazine*, pp. 96-115, December 1992.

[DIAM 96] J. E. Diamond and A. S. Alfa, "Analytical Models of Queues with Self-Similar Traffic", to appear in *Performance Evaluation* Journal, 1996.

[DOSH 96] B. T. Doshi et al., "A Broadband Multiple Access Protocol for STM, ATM, and Variable Length Data Services on Hybrid Fiber-Coax Networks", *IEEE 802.14 Working Group Document No. 802.14-96/222*, September, 1996.

[DUFF 93] N. G. Duffield and N. O'Connel, "Large Deviations and Overflow Probabilities for the General Single-Server Queue, with Applications". *Technical Report DIAS-STP-93-30*, Dublin Institute for Advanced Studies, 1993.

[ERRA 90] A. Erramilli and R. P. Singh, "The Application of Deterministic Chaotic Maps to Characterize Traffic in Broadband Packet Networks", *Proceedings 7th ITC Specialists Seminar*, 1990.

[ERRA 95] A. Erramilli and P. Pruthi, "Heavy-Tailed ON/OFF Source Behaviour and Self-Similar Traffic", *Proceedings of IEEE ICC '95*, pp. 445-450, Seattle, June 1995.

[GARR 94] M. W. Garrett and W. Willinger, "Analysis, Modelling and Generation of Self-Similar VBR Video Traffic", *Proceedings ACM SIGCOMM '94*, London, August 1994.

[GILL 95] S. E. Gillett, “Connecting Homes to the Internet: An Engineering Cost Model of Cable vs. ISDN”, *MIT Laboratory for Computer Science, Tech. Rep. 654*, June, 1995.

[GOLE 94] S. J. Golestani, “A Self-Clocked Fair Queueing Scheme for Broadband Applications”, *Proceedings of IEEE INFOCOM 94*, pp. 636-646, Toronto, 1994.

[GRAN 80] C. W. J. Granger and R. Joyeux, "An introduction to long-memory time series models and fractional differencing", *J. Yime Series Anal.* vol. 1, pp. 15-29, 1980.

[HEYM 82] D. P. Heyman, “An Analysis of the Carrier-Sense Multiple-Access Protocol”, *The Bell System Technical Journal*, vol. 61, no. 8, October 1982.

[HOEL 72] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Stochastic Processes*, Chapter 1, pp. 2-46, Houghton Mifflin, Boston, 1972.

[HOSK 81] J. R. M. Hosking, "Fractional differencing", *Biometrika*, vol. 68, pp. 165-176, 1981.

[HOSK 84] J. R. M. Hosking, "Modelling Persistence in Hydrological Time Series Using Fractional Differencing", *Water Resources Research*, vol. 20, no. 12, pp. 1898-1908, 1984.

[HOUC 95] D.J. Houck and W. S. Lai, “A Simulation Tool for Traffic Analysis of Hybrid Fibre-Coax Systems”, *Hybrid Fibre-Coax Systems Conference (Photonics East ’95), SPIE Proceedings vol. 2609*, pp. 243-250, Philadelphia, October, 1995.

[HOUC 96] D.J. Houck and W. S. Lai, “Traffic Analysis of Hybrid Fibre-Coax Systems”, *Broadband Access Systems Conference, SPIE Proceedings vol. 2917*, pp. 350-357, Boston, November, 1996.

[HUAN 95] C. Huang, M. Devetsikiotis, I. Lambadaris and A. Roger Kaye, "Fast Simulation for Self-Similar Traffic in ATM Networks", *Proceedings of IEEE ICC '95*, pp. 438-444, Seattle, June 1995.

[HURS 51] H. E. Hurst, "Long-Term Storage Capacity of Reservoirs", *Transactions of the American Society of Civil Engineering*, vol. 116, pp. 770-799, 1951.

[JAIN 86] R. Jain and S. A. Routhier, "Packet trains: Measurements and a new model for computer network traffic", *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 986-995, 1986.

[KARL 75] S. Karlin and H. M. Taylor, *A First Course in Stochastic Processes, 2nd Ed*, Chapter 4, pp. 117-166, Academic Press, London, 1975.

[KLEI 75] L. Kleinrock and F. A. Tobagi, “Packet Switching in Radio Channels: Part I - Carrier Sense Multiple Access Modes and Their Throughput-Delay Characteristics”, *IEEE Transactions on Communication*, 23, no. 12 (December 1975), pp. 1400-16.

[KLEI 76] L. Kleinrock, “Queueing Systems”, vol. 2, John Wiley & Sons, New York, 1976.

[LAUB 95] M. Laubach, “Proposal for System Simulation Performance Measurements - The General Model”, *IEEE 802.14 Working Group Document No. 802.14-95/151*, Montreal, October, 1995.

[LAVA 96] N. Lavaud and L. Decreusefond, "Simulation of Fractional Brownian Motion and Application to a Fluid Queue", *Proceedings of Australian Telecommunication Networks and Applications Conference ATNAC '96*, Melbourne, December, 1996.

[LELA 91] W. E. Leland and D.V. Wilson, "High Time Resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection", *Proceedings of IEEE INFOCOM '91*, pp. 1360-1366, Bal Harbour, 1991.

[LELA 94] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)", *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1-15, February 1994.

[LIKH 95] N. Likhanov, B. Tsybakov and N. D. Georganas, "Analysis of an ATM Buffer with Self-Similar("Fractal") Input Traffic", *Proceedings IEEE INFOCOM '95*, pp. 985 - 992, Boston, April 1995.

[LIMB 89] J.O. Limb, “Load-Controlled Scheduling of Traffic on High-Speed Metropolitan Area Networks”, *IEEE Transactions on Communications*, vol. 37, no. 11, November, 1989.

[LIMB 95] J.O. Limb and D. Sala, “An Access Protocol to Support Multimedia Traffic over Hybrid Fibre-Coax Systems”, *Proc. 2nd International Workshop in Community Networking*, pp. 35-40, Princeton, July, 1995.

[LIN 94] H.J. Lin and G. Campbell, “PDQRAP - Prioritised Distributed Queueing Random Access Protocol”, *Proceedings 19th Conference on Local Computer Networks*, pp. 82-91, Minneapolis, USA, October 1994.

[LIN 95] B. Lin and F. Koperda, “One Example to Evaluate a MAC Protocol for CATV Reverse Channel”, *IEEE 802.14 Working Group DocumentNo. 802.14/95-048*, May 1995.

[LIU 95] T. Liu, G. J. Anido and J. Chicaro, "ATM Traffic Modelling for Congestion Control", *Proceedings ATNAC '95*, pp. 91 - 96, Sydney, December 1995.

[MAN1 89] Proposed IEEE Standard 802.6 - Distributed Queue Dual Bus (DQDB) - Metropolitan Area Network (MAN), Draft D7, *IEEE 802.6 Working Group*, 1989.

[MAND 65] B. B. Mandelbrot, "Self-Similar Error Clusters in Communication Systems and the Concept of Conditional Stationarity", *IEEE Transactions on Communications Technology, COM-13*, pp. 71-90, 1965.

[MAND 68] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian motions, fractional noises and applications", *SIAM Rev.*, vol. 5, pp. 422-437, 1968.

[MAND 69] B. B. Mandelbrot, "Long-run linearity, locally Gaussian processes, H-spectra and infinite variances", *Intern. Econom. Rev.*, vol. 10, pp. 82-113, 1969.

[MAND 71] B. B. Mandelbrot, "A Fast Fractional Gaussian Noise Generator", *Water Resources Research*, vol.7, pp. 543-553, 1971.

[MAND 83] B. B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, New York, 1983.

[MATH 85] P. Mathys and P. Flajolet, “Q-ary Collision Resolution Algorithms in Random-Access Systems with Free and Blocked Channel Access”, *IEEE Trans. Inform. Theory*, vol. IT.31, pp. 217-243, March 1985.

[MAXW 96] K. Maxwell, “Asymmetric Digital Subscriber Line: Interim Technology for the Next Forty Years”, *IEEE Communications Magazine*, October 1996, Vol. 34 No. 10, pp. 100-108.

[MAXW 97] K. Maxwell, “Cable Modems and ADSL”, *The ADSL Forum Homepage at http://www.adsl.com/adsl_vs_cable.html.*

[MEIE 91] K. S. Meier-Hellstern, P. E. Wirth, Y. L. Yan and D. A. Hoeflin, "Traffic models for ISDN data users: Office Automation application", *Proceedings I.T.C. 13*, p. 167, Copenhagen, 1991.

[METC 76] R. M. Metcalfe and D. R. Boggs, “Ethernet: Distributed Packet Switching for Local Computer Networks”, *ACM Communications*, pp. 395-404, 1976.

[MIKH 79] V. A. Mikhailov, “Methods of Random Multiple Access*”, Candidate Engineering thesis*, Moscow Institute of Physics and Technology, Moscow, 1979.

[MOMO 95] M. Momona, “Framed Pipeline Polling for Cable TV Networks [Rev 2.0]”, *IEEE 802.14 Working Group Document No. 802.14-95/100*, September, 1995.

[MONT 95] S. P. Monti, "Broadband Data Services over Hybrid Fibre-Coax Networks", *Proceedings of the 5th ATM/B-ISDN Technical Workshop*, Gold Coast, Australia, June 1995.

[NAND 91] S. Nanda, D. J. Goodman, and U. Timor, “Performance of PRMA: A packet voice protocol for cellular systems”, *IEEE Transactions on Vehicular Technology*, vol. 40, no. 3, pp. 584-598, August 1991.

[NARA 97] P. Narasimhan et al., “Design and Performance Analysis of Radio Access Protocols in WATMnet, a Prototype Wireless ATM Network”, *Proceedings of 6th WINLAB Workshop on Third Generation Wireless Information Networks*, New Jersey, March, 1997.

[NEAM 95] T. Neame, R. Addie, M. Zukerman and F. Huebner, "Investigation of Traffic Models for High Speed Data Networks", *Proceedings ATNAC '95*, pp. 109 - 114, Sydney, December 1995.

[NORR 94] I. Norros, "A storage model with self-similar input", *Queueing Systems*, 16:387-396, 1994.

[NORR 95] I. Norros, "The Management of Large Flows of Connectionless Traffic on the Basis of Self-Similar Modelling", *Proceedings of IEEE ICC '95*, pp. 451-455, Seattle, June 1995.

[OAKL 91] K. A. Oakley et al., "Fibre in the Access Network", *British Telecommunications Engineering*, vol. 10, pp. 40-47, April, 1991.

[O'NEI 92] C. O'Neill, "Fairness Discarding for Congestion Control in ATM Networks", *Proceedings of Australian Broadband Switching and Services Symposium '92*, Vol. 1, pp. 185-192, Melbourne, July 1992.

[PANC 95] P. Pancha and M. Karol, "Guaranteeing Bandwidth and Minimizing Delay in Packet Switched (ATM) Networks", *Proceedings of IEEE Globecom 95*, pp. 1064-1070, Singapore, November, 1995.

[PASS 97] N. Passas, S. Paskalis, D. Vali, and L. Merakos, "Quality of Service Oriented Medium Access Control for Wireless ATM Networks", *IEEE Communications Magazine*, vol. 35, no. 11, November, 1997.

[PITT 93] J. M. Pitts et al., " A Techno-Economic Comparison of the RACE 2024 BAF System and Other Possible Optical Access Systems", *Proceedings of the RACE Open Workshop on Broadband Access*, Nijmegen, The Netherlands, June, 1993.

[QURE 95] A. G. Qureshi and G. J. Anido, "Modelling Long-Range Dependence in ATM VBR Video Traffic", *Proceedings ATNAC '95*, pp. 103 - 108, Sydney, December 1995.

[RAMA 88] K. K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with a Connectionless Network Layer", *Proceedings of SIGCOMM '88*, California, August 1988.

[RAYC 92] D. Raychaudhuri, N. D. Wilson, "ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks", *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 8, October 1992.

[RAYC 97] D. Raychaudhuri et al., "WATMnet: A Prototype Wireless ATM System for Multimedia Personal Communication", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 1, January, 1997.

[RIVE 85] R. L. Rivest, "Network Control by Bayessian Broadcast", *MIT Report no. LCS-TM-285*, Massachussets Institute of Technology - Laboratory for Computer Science, Cambridge, MA, 1985.

[ROBE 72] L. G. Roberts, "Aloha Packet System with and without Slots and Capture" ASS Note 8, Stanford Research Institute, Advanced Research Projects Agency, Stanford, CA, 1972.

[ROY 93] M. F. Roy, "Who is Building Tomorrow's Network", *Proceedings of the RACE Open Workshop on Broadband Access*, Nijmegen, The Netherlands, June, 1993.

[SALA 95] D. Sala, "MAC Protocols for Multimedia Data over HFC Architecture", *Georgia Tech Technical report GIT-CC-95-48, October 1995.*

[SALA 96a] D. Sala and J. O. Limb, "A Protocol for Efficient Transfer of Data over FIber/Cable Systems", *Proceedings of IEEE INFOCOM '96*, pp. 904-910, San Francisco, March 1996.

[SALA 96b] D. Sala and J. O. Limb, "Scheduling Disciplines for HFC Systems: What can we learn from ATM Scheduling?", *Proc. 3rd International Workshop in Community Networking*, pp. 13-18, Antwerpen (Belgium), May 23-24, 1996.

[SALA 96c] D. Sala, D. Hartman, and J. O. Limb, "Comparison of Algorithms for Station Registration on Power-up in an HFC Network", *IEEE 802.14 Working Group Document No. 802.14-96/012*, January, 1996.

[SLOS 94] R. Slosiar, "Busy and Idle Periods at an ATM Multiplexer Output Resulting from the Superposition of Homogenous ON/OFF Sources", *Proceedings of the 14th International Teletraffic Congress*, Vol 1a., pp. 431-440, Antibes, France, June, 1994.

[SLOS 96] R. Slosiar, "A Fair and Efficient Minimal Cell Delay Variation Access Network", *Proceedings of the 8th IEEE LAN/MAN Workshop*, Berlin-Potsdam, August, 1996.

[SRIR 95] K. Sriram, "An Adaptive Digital Access Protocol (ADAPt) for Multiservice Broadband Access Networks (Part I): Protocol Description", *IEEE 802.14 Document No. 802.14-95/046*, June 1995.

[TAHA 76] H. A. Taha, *Operations Research: An Introduction*, Chapter 13, pp. 450-500, Collier Macmillan International Editions, London, 1976.

[TAQQ 86] M. S. Taqqu and J. B. Levy, "Using renewal processes to generate long-range dependence and high variability," *in Dependence in Probability and Statistics, E. Eberlein and M. S. Taqqu, eds.* Boston, MA: Birkhauser, vol. 11, pp. 73-89, 1986.

[TAQQ 95] M. S. Taqqu, W. Willinger and V. Teverovsky, "Estimators for long-range dependence: an empirical study", *Fractals*, vol. 3 no. 4, pp. 785-788, 1995.

[TSYB 85] B. Tsybakov, "Survey of USSR Contributions to Random Multiple-Access Communications", *IEEE Transactions on Information Theory*, vol. 31, no.3, pp. 143-165, 1985.

[TSYB 96] B. Tsybakov and N. Georganas, "On Self-Similar Traffic in ATM Queues: Definition, Overflow Probability Bound and Cell Delay Distribution", *submitted for publication in IEEE/ACM Transactions on Networking*.

[ULM 95] J.M. Ulm, C. Grobicki, "UniLINK as a Media Access Protocol for Community Cable TV", *Second International Workshop in Community Networking*, pp. 41-48, Princeton, July, 1995.

[VASS 92] S. Vassilopoulos and P. Papantoni-Kazakos, "A Transmission Scheduling Algorithm for Mixed Traffic: High and Low Priority", *INFOCOM '92*, pp. 2251-2259, Italy, May 1992.

[WAND 97] The Magic WAND Wireless ATM Demonstrator, *URL: http://www.tik.ee.ethz.ch/~wand*, 1997.

[WARI 91] D. L. Waring, "The Asymmetrical Digital Subscriber Line (ADSL): A New Transport Technology for Delivering Wideband Capabilities to the Residence", *Proceedings of IEEE Globecom '91*, pp. 1979-1986, December 1991.

[WHIT 96] R. Whittle, "The Optus Vision: Telephony, Internet and Video", Australian Communications Magazine, pp. 61-77, August 1996.

[WU 94] C. Wu, G. Campbell, "Extended DQRAP (XDQRAP). A Cable TV Protocol Functioning as a Distributed Switch", *Proc. 1st International Workshop on Community Networking Integrated Multimedia*, pp. 191-198, San Francisco, USA, July 1994.

[XIE 95] H. Xie, R. Yuan and D. Raychaudhuri, "Data Link Control Protocols for Wireless ATM Access Channels", *Proceedings of ICUPC '95*, Tokyo, November, 1995.

[XU 93] W. Xu and G. Campbell, “A Distributed Queueing Random Access Protocol for a Broadcast Channel”, *Proc. SIGCOMM ’93*, pp. 270-278, Ithaca, New York, 1993.

[ZHAN 95] H. Zhang, “Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks”, *Proc. IEEE Globecom ‘95*, pp. 1064-1070, Singapore, November 1995.

[ZUKE 86] M. Zukerman and I. Rubin, "On Multi Channel Queueing Systems with Fluctuating Parameters," *Proceedings of IEEE INFOCOM '86*, pp. 600-608, Miami, Florida, April 1986.

[ZUKE 90] M. Zukerman and P. G. Potter, "The DQDB Protocol and its Performance under Overload Traffic Conditions", *Computer Networks and ISDN Systems*, vol. 20, pp. 261-270,1990.

[ZUKE 94] M. Zukerman and S. Chan, “Congestion Control by Maintaining Fairness in High Speed Data Networks”, *Proceedings of IEEE Globecom ’94*, pp. 1576-1580, San Francisco, November-December, 1994.