An Investigation of Static and Dynamic Call Admission Control Methods By Simulation

A project submitted to the

Department of Mathematics and Statistics

in conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

September, 1997

Abstract

Fast data rates and quality of service guarantees in future ATM-based Broadband Integrated Services Digital Networks require Call Admission Control (CAC) to prevent network congestion and maintain satisfactory performance. A lot of research has been done in developing and extending effective bandwidths to more and more kinds of sources. The lack of simple VBR source models and the presence of long-range dependence make effective bandwidths increasingly complex to determine. Measurement-based CAC methods do not require a model of the source, and can adapt to errors in declared traffic descriptors.

This project investigates peak-rate only Dynamic CAC. Three parameters are necessary in the method, and they are found to have a large impact on link utilization and cell loss. Dynamic CAC and static CAC (involving effective bandwidths) are compared using simulated On-Off sources. With the proper choice of parameters Dynamic CAC will admit more sources while maintaining zero cell loss. When the sustained cell rate is overspecified, performance under effective bandwidth CAC suffers, while Dynamic CAC is unaffected. With a VBR video trace that exhibits long-range dependence, Dynamic CAC is effective, even when a traffic model or effective bandwidth are unknown.

Acknowledgments

I would like to thank Dr. H. Mouftah for suggesting CAC as an interesting area for investigation.

I am grateful to Dr. F. Alajaji for allowing me to use his computers when I needed more power.

I am most grateful to Dr. G. Takahara for his guidance, support and patience. I would also like to express appreciation for all the resources he has made available to me. Without them this project would not have been possible.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

1.1 Asynchronous Transfer Mode (ATM)

1.1.1 ATM Networks

Asynchronous transfer mode (ATM) will be used for future Broadband Integrated Services Digital Networks (B-ISDN). In ATM a source is connected to its destination via a *virtual channel* (VC). A source generates a stream of packets, called *cells*, which are always 53 bytes long, including a 5 byte header. An aggregate of VCs is called a *virtual path* (VP). Virtual channels for different source-destination pairs may share the same physical link. Thus the potential exists for some connections to degrade the performance of others. In this context performance measures are typically delay or cell loss probability, but may include others like bit error probability.

Several types of services have been defined for ATM ([2], [3]). Constant Bit Rate (CBR) connections allocate a constant amount of bandwidth for a connection. Variable Bit Rate (VBR) connections are for sources whose bandwidth requirement varies. Instead of allocating a constant rate based on the sources highest demand case, VBR connections attempt to use resources more efficiently. This can be accomplished by taking advantage of statistical multiplexing (described below). VBR connections are further subdivided into real-time (rt-VBR) for services with tight constraints on delay and delay variation, and non-real-time (nrt-VBR) for services without such tight constraints. Available Bit Rate (ABR) connections guarantee a minimum bandwidth (which may be requested as zero). An elastic component above this minimum may be used, with feedback control allowing dynamic adjustment of the source's sending rate. Unspecified Bit Rate (UBR) connections are also possible. The ITU-T B-ISDN specification of 'ATM Transfer Capability' (I.371) parallels those of the ATM Forum in respects. Substantial differences are that rt-VBR services are left for further study, there is no analogue to UBR services, and a unique service called ATM Block Transfer (ABT) is specified [3].

An important aspect of ATM is the quality of service (QoS) guarantee. Each source requests a connection by providing traffic descriptors (Table 1.1). The Peak Cell Rate (PCR) is defined as the inverse of the minimum interarrival time between requests to transmit cells. For VBR services the Sustainable Cell Rate (SCR) may be optionally specified. This is an upper bound on the average cell rate. If SCR is specified the Burst Tolerance (τ_s) must also be specified. These two parameters are related by the Generic Cell Rate Algorithm (GCRA) which may be interpreted as a leaky bucket regulator. For ABR services the Minimum Cell Rate (MCR) will be provided to distinguish between the fixed and elastic regions of requested bandwidth. A value for the Cell Delay Variation Tolerance (τ) may be selected by the user. This represents the tolerance for the jitter in cell delay variation introduced by the network. Although selected by the user it is not a characteristic of the source.

Traffic Descriptor	Type of Service
Peak Cell Rate (PCR)	CBR, VBR, ABR
Sustainable Cell Rate (SCR)	VBR (optional)
Burst Tolerance (τ_s)	
Minimum Cell Rate (MCR)	ABR
Cell Delay Variation Tolerance (τ)	CBR, VBR

Table 1.1: ATM Traffic Descriptors

Along with traffic parameters, a connection request includes QoS parameters (cell loss ratio (CLR), and possibly cell transfer delay (CTD) and cell delay variation (CDV)). Typical values for CLR are 10^{-7} to 10^{-9}. Based on the ATM traffic descriptors, the network determines whether a connection can be established that guarantees performance at least as good as specified in the QoS parameters. A connection is established only if such a guarantee can be made. Rejection may lead to re-negotiation by the source with less stringent requirements.

A key component in an ATM network system is the ATM multiplexer (Figure 1.1). The ATM multiplexer is simply a finite buffer connected to many inbound VCs and one outbound high-speed link. Cells arrive through the VCs, attempting to fill the buffer. The high speed link serves to drain the buffer as cells are transmitted on the high-speed link. The buffer limits the effect of *cell scale congestion* and *burst scale congestion* [44]. Cell scale congestion is caused by the simultaneous arrival of cells from many VCs. Only a small buffer is needed to prevent cell loss. Burst scale congestion is caused by the aggregate data rate across all VCs exceeding the link capacity for a short time. Larger buffers reduce the probability of loss, but increase delay. If the buffer is full arriving cells are lost. A large number of cells within the buffer will mean that cells experience delay before being transmitted on the high-speed link. Thus, to meet the QoS requirements of established VCs the multiplexer may

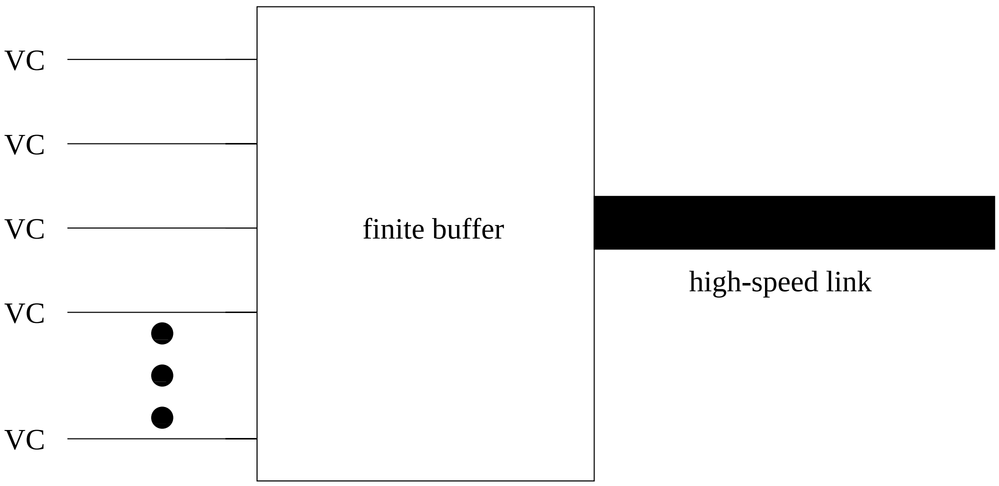

Figure 1.1: An ATM Multiplexer

have to prevent a new requested connection from being established. This is the call admission control (CAC) function of ATM networks and is a preventative approach to avoiding network congestion and maintaining network performance. Admission policies may use sets of curves or real-time calculations as the basis for decision-making. Because of the fast data rates involved real-time calculations cannot be exceedingly complex.

1.1.2 Statistical Multiplexing

One can associate with a VBR source a probability distribution of its bit rate, $p(x)$, where x is the bit rate ([9],[23]). The probability distribution of N superposed sources is given by

$$\tilde{p}(x) = p_1 \star p_2 \star \cdots \star p_N(x) \tag{1.1}$$

where $\star$ denotes convolution. Statistical multiplexing takes advantage of the idea that the tail of $\tilde{p}(x)$, where most of the sources are sending at a high rate, has low probability, on the conditions that enough sources are multiplexed and they are not correlated.

Unfortunately, for the purposes of CAC, the complete distribution of a source is usually unknown, and convolutions are often impractical. Although currently unsupported by ATM [2], knowing the variance as well as the mean rate allows approximation of $\tilde{p}(x)$ by a Gaussian distribution. With a large enough number of sources the approximation may be sufficiently accurate to be useful.

1.2 Outline

This project presents results from simulations comparing effective bandwidth-based CAC with Dynamic CAC. In chapter 2 some background into effective bandwidths and the Dynamic CAC algorithm are presented. Issues in the modelling of VBR video sources, and their impact on CAC are also discussed. In chapter 3 Dynamic CAC is investigated to find useful parameter values for simulated On-Off sources. Static CAC (primarily effective bandwidths) and Dynamic CAC are then compared under the assumption that the ATM sustained cell rate is declared perfectly, or overspecified. In chapter 4 a VBR video trace, known to be long-range dependent, is used to investigate the performance of Dynamic CAC for long-range dependent traffic when an appropriate model may not be known.

Chapter 2

Literature Review

2.1 Call Admission Control (CAC)

In the literature call admission control can be considered in three general areas: static CAC ([11], [13], [17], [23], [24], [27], [31], [38], [41]), measurement-based CAC ([18], [34], [45], [46], [47], [48]), and fuzzy-logic/neural-network based CAC ([10],[37]). The simplicity of static CAC is perhaps the main reason it has generated a good deal of interest, despite its limitations. Measurement-based CAC offers advantages through its robustness, although it may be more complex. Fuzzy-logic and neural network based methods represent a different technique (for references see [10]).

While cell loss and delay are both important measures of performance, it is often possible to address cell delay through control of buffer size. Indeed, for a first-in first-out (FIFO) buffer of size K cells and a high speed link with rate R bits per second (bps) the delay, D, in seconds due to queueing in the buffer is upper bounded by:

$$D \leq \frac{424K}{R} \tag{2.1}$$

The probability of cell loss due to buffer overflow is often the quantity of interest when studying CAC techniques. Dealing with link delay due to queueing in this way means both types of VBR traffic can be handled together.

2.2 Static CAC

Static CAC methods use the declared trafic descriptor with no attempt to measure the actual traffic. They assume the declared values are good descriptions of the traffic. Peak-rate based and effective bandwidths are two well-known forms. Mean-rate based (or SCR-based) CAC is also possible but is expected to have higher cell loss than effective bandwidth-based CAC for reasonable traffic sources and buffer sizes. Logarithm and simplified logarithm CAC [40] are also known.

2.2.1 Peak Rates

Perhaps the simplest form of static CAC is peak-rate based admission. For N existing connections with peak rates R^i_{peak} $(i = 1, \ldots, N)$, requested connection with peak rate R^{N+1}_{peak}, and link capacity C, this policy checks:

$$\sum_{i=1}^{N} R^i_{peak} + R^{N+1}_{peak} \leq C \tag{2.2}$$

The inequality must be satisfied to accepts the connection. Such a policy avoids burst-scale congestion effects, thus allowing small buffer sizes. By ignoring statistical multiplexing effects it is quite conservative.

2.2.2 Effective Bandwidths

As opposed to peak-rate based CAC, resources can be used more efficiently through the use of effective bandwidths. (*e.g.* [16], [26], [17], [11], [27], [7]) Put simply, the *effective bandwidth* of a source is the least amount of bandwidth $\hat{c}$ required by a source, for a given buffer size, that ensures the probability of buffer overflow remains smaller than some desired value (ε). Intuitively, $\hat{c}$ will be some number between the source's mean rate (R_{mean}) and its peak rate (R_{peak}).

As described in [17] a single two-state Markov source can be completely determined by its peak rate R_{peak}, utilization ρ, and mean burst period b. Then, for a finite buffer of size x, link capacity C and desired overflow probability ε the following relation holds:

$$\varepsilon = \beta \times \exp\left(\frac{-x\,(c - \rho R_{peak})}{b(1-\rho)(R_{peak} - c)c}\right) \tag{2.3}$$

where

$$\beta = \frac{(c - \rho R_{peak}) + \varepsilon\rho\,(R_{peak} - c)}{(1-\rho)c} \tag{2.4}$$

(For an infinite buffer the general equation (2.3) is the same, but β will have a different value.) The effective bandwidth for the single two-state source can be found by numerically solving 2.3.

A simplification is to take $\beta = 1$ (Note: $\beta \leq 1$). (In [17] it is stated that extensive numerical experiments never produced a relative error greater than 10%.) This allows an explicit formula for the effective bandwidth:

$$\hat{c} \approx \frac{\alpha b(1-\rho)R_{peak} - x + \sqrt{[\alpha b(1-\rho)R_{peak} - x]^2 + 4x\alpha b\rho(1-\rho)R_{peak}}}{2\alpha b(1-\rho)} \tag{2.5}$$

where $\alpha = \ln \frac{1}{\varepsilon}$.

In [1] an ON-OFF Markov fluid model is used to represent homogeneous ON-OFF sources. In this model a source switches between the ON (burst) state and the OFF (idle) state. In the ON state data is sent at the source's peak rate. In the OFF state no data is sent. In [1] the length of both ON and OFF periods are exponentially distributed with mean 1 and $\frac{1}{\lambda}$ respectively. The unit of time is taken to be the average ON period and the unit of information is the amount of data in one time unit. The link speed is C. In this model, when N sources are in the ON state they transmit N units of information. It is convenient to assume the buffer is infinite and a stability condition equivalent to the following is satisfied:

$$\frac{1}{C}\left(\frac{N\lambda}{1+\lambda}\right) < 1 \tag{2.6}$$

The entire left-hand side of (2.6) represents the traffic intensity and the term in brackets is simply the average incoming rate due to N sources.

$$\text{Let } G(x) \quad := \quad \Pr(\text{buffer content} > \text{x})$$

For the negative (or *stable*) eigenvalues $z_{N-\lfloor C\rfloor-1} < \cdots < z_1 < z_0$, and associated eigenvectors ϕ_i, of a certain set of differential equations,

$$G(x) = -\sum_{i=0}^{N-\lfloor C \rfloor -1} e^{z_i x} a_i([1, 1, \ldots, 1]\, \phi_i) \tag{2.7}$$

where

$$a_j = -\left(\frac{\lambda}{1+\lambda}\right)^N \prod_{i=0,\, i\neq j}^{N-\lfloor C \rfloor -1} \frac{z_i}{z_i - z_j}, \quad 0 \leq j \leq N - \lfloor C \rfloor - 1$$

Generally, the probability of overflow beyond x is of interest for x relatively large. Thus, an asymptotic result for $G(x)$, which is easier to compute, is useful. Noting that (2.7) is the sum of exponential terms, the term for z_0 will dominate. A convenient formula for z_0 is:

$$z_0 = -\frac{1 + \lambda - N\frac{\lambda}{C}}{1 - \frac{C}{N}} = -\frac{(1+\lambda)(1-\rho)}{1 - \frac{C}{N}} \tag{2.8}$$

yielding

$$G(x) \approx \rho^N \left(\prod_{i=1}^{N-\lfloor C \rfloor -1} \frac{z_i}{z_i - z_0} \right) e^{z_0 x} \tag{2.9}$$

In the more suggestive notation of [17] this becomes

$$G(x) \approx \left(\frac{N\rho R_{peak}}{C}\right)^N \left(\prod_{i=1}^{N-\lfloor \frac{C}{R_{peak}} \rfloor -1} \frac{z_i}{z_i - z_0} \right) \exp\left(-\frac{N(C - N\rho R_{peak})}{b(1-\rho)(NR_{peak} - C)C} x \right) \tag{2.10}$$

where ρ is now understood to be the utilization for a single source.

Typically, for N superposed sources, results for $G(x)$ must be found numerically, and approximations to an infinite buffer using the dominant eigenvalue are used. In [28] results involving $G(x)$ are generalized to sources with ON times whose mean is some constant, whose distribution is non-exponential (*e.g.* Erlang, hyperexponential), and which are not homogeneous. The approximation

$$G(x) \sim \beta e^{z_0 x} \tag{2.11}$$

is used. However, unlike the results in [1], which yield an expression for β, determining it is extremely difficult. The approximation $\beta = 1$ can sometimes be used to advantage, which gives

$$G(x) \sim e^{z_0 x} \tag{2.12}$$

For a desired cell loss probability (ε) we want $G(x) \leq \varepsilon$ and can take

$$z_0 = \frac{ln\varepsilon}{x} = -\frac{\alpha}{x} \tag{2.13}$$

where $\alpha = \ln \frac{1}{\varepsilon}$.

In [28] it is shown that for non-homogeneous ON-OFF sources whose states are exponentially distributed with mean $b_i = \frac{1}{\mu_i}$ and $\frac{1}{\lambda_i}$ respectively, and peak rate γ_i, the required capacity (effective bandwidth) for N superposed sources ($\hat{C}$) is

$$\hat{C} = \frac{\sum_{i=1}^{N} \left((\lambda_i + \mu_i + \gamma_i z_0) - \sqrt{(\lambda_i + \mu_i + \gamma_i z_0)^2 - 4\lambda_i \gamma_i z_0} \right)}{2z_0} \tag{2.14}$$

By combining (2.13), (2.14), and the relations

$$\mu_i = \frac{1}{b_i} \ , \ \lambda_i = \frac{\rho_i}{b_i(1-\rho_i)} \ , \ \lambda_i + \mu_i = \frac{1}{b_i(1-\rho_i)} \tag{2.15}$$

one can derive [17] the simple relationship between the effective bandwidth of the individual sources, $\hat{C}_i$, as in (2.5), and the aggregate, $\hat{C}$:

$$\hat{C} = \sum_{i=1}^{N} \hat{C}_i \tag{2.16}$$

Thus, the static CAC admission policy, using effective bandwidths for N existing connections, and the $N+1$ requested connection, checks

$$\sum_{i=1}^{N} \hat{C}_i + \hat{C}_{N+1} \leq C \qquad (2.17)$$

where C is the capacity of the high-speed link out of the buffer. The new connection will only be permitted if the above inequality is satisfied.

In [11] effective bandwidths are extended to Markov sources with more than two states (*e.g.* discrete autoregressive models of order one, DAR(1), used to model teleconference traffic [20]) and Markov Modulated Poisson Processes (MMPPs).

In [27], large deviation techniques are used to generalize effective bandwidths to multi-state discrete time Markov sources, multi-state Markov fluids, and Markov Modulated Poisson Processes. The linearity condition (2.16) extends to all of these kinds of sources so the CAC policy (2.17) extends in the obvious way.

2.2.3 Problems with Effective Bandwidths

Effective bandwidths are usually considered to be over-conservative [17] but sometimes can be the opposite [7]. In Guérin *et al.* [17] the approximation $\beta = 1$ is discussed. In particular, β is sensitive to the number of sources and the mean burst length. When either of these is large β may be many orders of magnitude smaller than 1. A suggested remedy is to approximate the aggregate stationary bit rate with a Gaussian distribution and look at the tail of the distribution.

For an aggregate bit rate with mean $m = \sum_{i=1}^{N} m_i$, variance $\sigma^2 = \sum_{i=1}^{N} \sigma_i^2$, and desired cell loss probability ε:

$$\hat{C}_{Gauss} \approx m + \sigma\sqrt{-2\ln(\varepsilon) - \ln(2\pi)} \tag{2.18}$$

For N ON-OFF sources with peak rate, R_{peak}, and utilization, ρ, this becomes

$$\hat{C}_{Gauss} \approx N\rho R_{peak} + R_{peak}\sqrt{N\rho(1-\rho)}\sqrt{-2\ln(\varepsilon) - \ln 2(\pi)} \tag{2.19}$$

As in (2.16)

$$\hat{C}_{Flow} = \sum_{i=1}^{N} \hat{C}_i \tag{2.20}$$

Both are conservative, over a different range of parameters, so the effective bandwidth of the aggregate can be taken as

$$\hat{C} = \min\{\hat{C}_{Flow}, \hat{C}_{Gauss}\} \tag{2.21}$$

The Gaussian assumption does not hold for small numbers of very bursty connections with high peak rates, low utilizations and long burst periods. In these cases allocating the bandwidth computed under the flow model as a *constant* bit rate connection is recommended.

In [7] numerical techniques are used to show that effective bandwidths from approximation (2.12) get worse as the number of sources increases, the buffer size decreases, the channel utilization decreases, the target blocking probability increases, and the sources get further from Poisson. They also give an example where the effective bandwidths are not conservative. Their situation applies to traffic streams in which cells arrive in a manner less bursty than Poisson and small buffer sizes (less than 20 cells). This serves as a reminder that the approximation is for relatively large buffer sizes.

A weakness of effective bandwidth techniques for all source models is that all the required values are not currently provided as ATM traffic descriptors. Even in the simplest case of ON-OFF sources with exponential states the mean burst length is a required value that is not provided. More complex Markov models require even more additional values. The same holds for the Gaussian approximation improvement which requires the variance of each source.

While there has been success modelling video-teleconference traffic [20], a single model for video sources seems far off, if not impossible. Multiple models require multiple effective bandwidth formulas. This would seem to increase the complexity of any CAC algorithm using effective bandwidths.

2.3 Measurement-Based CAC

Static CAC policies assume that existing connections closely follow declared traffic descriptor values. Yet, values such as the sustainable cell rate, which is upper bound of the average cell rate, are difficult to measure exactly. The value declared may be an overestimate. As well, a policy that uses the sustained cell rate in place of the true mean rate has a built-in error. Assumptions that sources are independent may be inaccurate. Additional problems include lack of an accurate reference model for the cell arrival process and the possibility that the traffic exceeds its declared traffic descriptors. CAC policies that measure the incoming traffic attempt to avoid or limit these difficulties ([45], [46], [48], [43], [34], [18].)

2.3.1 Dynamic CAC

In [45] 'Dynamic' CAC is proposed, which uses the declared peak and average rates for a connection as well as measurements of the number of cells arriving. A simplified version which does not use the average rate is described in [48].

Throughout, the time unit is take to be the time required to place one cell on the link. Cell delay is addressed by buffer engineering as described above (2.1). Dynamic CAC defines a *measurement period* of length r (cells) in which the number of cells arriving at the output buffer are counted. Let x_i be the number of cells counted during the ith measurement period. Let

$$q(k) = \frac{1}{M} \sum_{i=1}^{M} \begin{cases} 1 & , \quad \text{if } x_i = k \\ 0 & , \quad \text{otherwise} \end{cases} \quad , \quad k = 0, 1, \ldots \tag{2.22}$$

be calculated for each M measurement periods. Then $\{q(k)\}$ is the observed probability distribution of the number of cells arriving at the buffer each measurement period.

Let $\{p(k)\}$ be the actual probability distribution of the number of cells arriving during a measurement period. The cell loss ratio (CLR) is upper-bounded by [47]:

$$CLR \leq \frac{\sum_{k=r}^{\infty} (k-r)p(k)}{\sum_{i=0}^{\infty} ip(i)} \tag{2.23}$$

where $r < \frac{K+1}{2}$ for a stationary cell arrival process and $r < K + 1$ for an ergodic stationary cell arrival process.

In Dynamic CAC $\{p(k)\}$ is estimated by $\{\hat{p}(k)\}$, which is updated every M measurement periods with the information in $\{q(k)\}$. Thus, these M measurement periods can be referred to as a *renewal period*. A renewal period is M measurement periods, or Mr time units. The new estimate of $\{p(k)\}$ is given by:

$$\hat{p}_{new}(k) = \alpha q(k) + (1 - \alpha)\hat{p}_{old}(k) \quad , \quad k = 0, 1, \ldots \tag{2.24}$$

where α is a constant such that $0 \leq \alpha \leq 1$. This constant exponentially controls the persistence of old information.

Consider a new source requesting a connection. Let R be the number of cells it could send at its declared peak rate in a measurement period (*i.e.* $R = \lceil \frac{R_{peak} \times r}{C_{link}} \rceil$).

Let $p_{N+1}(k)$ be the actual (but unknown) probability distribution for the new source. Ideally one would like to calculate:

$$\tilde{p}(k) = p \star p_{N+1}(k) \tag{2.25}$$

and then

$$CLR \leq \frac{\sum_{k=r}^{\infty}(k-r)\tilde{p}(k)}{\sum_{i=0}^{\infty} i\tilde{p}(i)} \tag{2.26}$$

Since the actual distributions are not knowable the convolution is replaced by:

$$\hat{\tilde{p}}(k) = \begin{cases} 0 & , \quad k < R \\ \hat{p}(k-R) & , \quad k \geq R \end{cases} \tag{2.27}$$

and the (estimated) upper-bound

$$CLR \leq \frac{\sum_{k=r}^{\infty}(k-r)\hat{\tilde{p}}(k)}{\sum_{i=0}^{\infty} i\hat{\tilde{p}}(i)} \tag{2.28}$$

is used. The call is admitted if the estimate is less than the cell loss objective associated with the multiplexer (Figure 2.1) under the network management scheme. (These are discussed in [35] and [48], for example.) This objective must be at least as stringent as the requested QoS. If the call is admitted $\{\hat{p}(k)\}$ is replaced by $\{\hat{\tilde{p}}(k)\}$. Under reasonable conditions this is initially a gross overestimate of the new actual probability distribution. The assumption that the new source will always send at its peak rate would typically be too severe. With time, updating the distribution will revise the estimated distribution downwards. When a call is finished no action is taken since updating will observe any changes this might cause in the number of cells arriving. The accuracy of $\{\hat{p}(k)\}$ in estimating is $\{p(k)\}$ is unclear and more research in this area is needed.

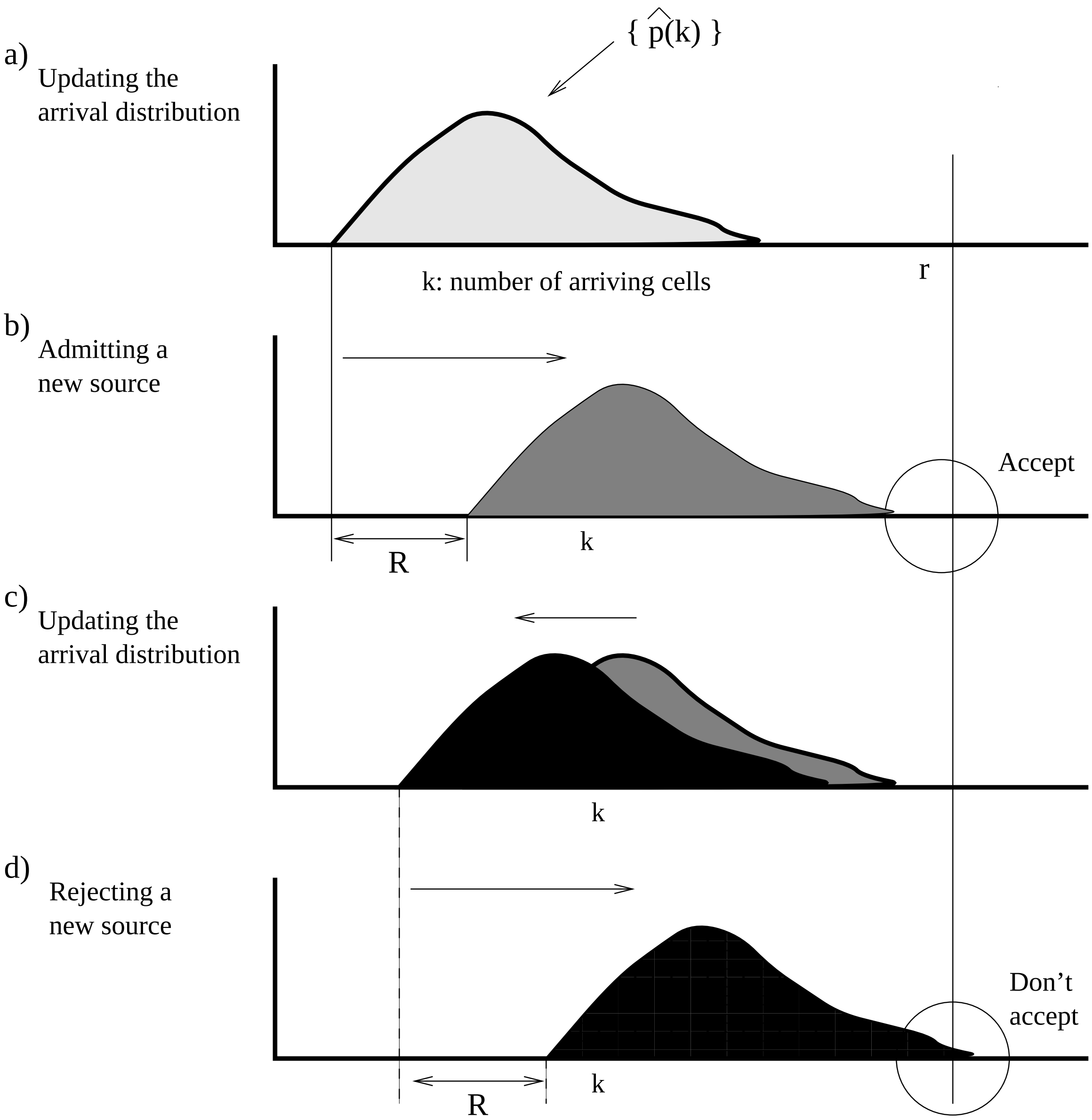

Figure 2.1: Illustration of Dynamic CAC

2.4 Traffic Models

To simulate VBR traffic sources and investigate multiplexer performance accurate traffic models are important. A variety of models have been developed for different kinds of traffic [25], [19], [20], [21], [36], [12].

2.4.1 Voice and Data Sources

One of the simplest VBR source models is the On-Off model for burst and idle periods. This model is used in [1] with little reference to real world applicability. Human speech with silence supression can be modelled this way. In [19] exponenetially distributed burst and idle periods with mean 352 ms and 650 ms, respectively, are given as a voice source model of this kind. The superposition of many voice sources is more complex. Various models have been proposed, including a Markov Modulated Poisson Process(MMPP) [19]. Data sources, for such things as file transfers, have also been modelled with On-Off sources.

2.4.2 Video Sources

Modelling of VBR video sources has proven difficult so far. Video teleconference, movies, news, sports and even TV bingo can fall into this category. Videoteleconference modelling has proven somewhat more tractable because of stability within the image. Other video sources have scene changes which increase the amount of data transferred, and thus, the bit rate. It has been reported that the autocorrelation structure between the number of cells per frame of video teleconference and other video sources is very different (see [21].)

In [36] a 10 second video teleconference trace is used to develop two models. The first model is a continuous-state autoregressive model for the bit rate $\lambda(n)$ during frame n. They found

$$\lambda(n) = 0.8781\lambda(n-1) + 0.1108w(n) \text{ where } w(n) \sim \mathrm{N}(0.572, 1) \tag{2.29}$$

Their second model, which is more useful for queueing analyses, uses a discrete-state continuous-time Markov process. The bit rate $\bar{\lambda}(t)$ of a source in continuous time is quantized into finite levels. The aggregate of N sources has bit rate $\bar{\lambda}_N(t)$. In steady-state it can be modelled with a binomial distribution. They further describe how $\bar{\lambda}_N(t)$ can be considered as the aggregate of M independent On-Off 'minisources', where M is found through parameter matching.

In [20] and [12] a Discrete Autoregressive model of order 1, or DAR(1), is used to model video teleconference traffic for several coding methods. Their technique models the process which counts the number of cells per frame and requires the peak rate, mean (m_0), variance (v_0) and the first-order autocorrelation coefficient. The process is governed by the transition matrix:

$$\mathbf{P} = \rho\mathbf{I} + (1-\rho)\mathbf{Q} \tag{2.30}$$

where ρ is the autocorrelation coefficient, $\mathbf{I}$ is the identity matrix, and each row of $\mathbf{Q}$

consists of negative binomial probabilities $(f_0, f_1, \ldots, f_K, F_K^c))$ where

$$f_k = \binom{k+r-1}{k} p^r (1-p)^k = \binom{-r}{k} p^r (p-1)^k \quad , \quad k = 0, 1 \ldots, \quad 0 < p < 1, \quad r > 0$$

and

$$F_K^c = \sum_{k>K} f_k$$

where K is the peak rate in cells per frame. The parameters p and r can be estimated by

$$\hat{p} = \frac{m_0}{v_0} \quad , \quad \hat{r} = \frac{m_0^2}{v_0 - m_0} \tag{2.31}$$

for m_0 and v_0 as described above. A technique is given to find the dominant eigenvalue of the multiplexing system with K homogeneous sources. This allows analysis of the buffer content, which is compared favourably with simulation results.

In [21] similar techniques are applied with mixed success to a variety of video traces including film, football and bingo. They look at the structure of the sources to find the number of cells in scene change frames, the number of cells per frame for frames within a scene and scene lengths. A variety of distributions including Pareto, Gamma, and Weibull are necessary to model some of these quantities. They conclude that the '... use of a single model based on a few physically meaningful parameters and applicable to all sequences does not seem to be possible.' An alternative may be to model sources with a Markov chain with many states, requiring many parameters.

2.4.3 Long-Range Dependence and Self-Similarity

In parallel with investigations of source modelling have been results involving long-range dependence of traffic sources ([14], [33], [42], [5], [22]). Discussion was generated when large traces of Ethernet traffic were shown to exhibit self-similarity (summarized in [33]). In [42] some features of long-range dependent and self-similar processes, are summarized.

Let A_t denote the amount of traffic (*e.g.* bits) offered in time interval $[0, t]$.

Let $A(s,t) = A_t - A_s$ denote the amount of traffic offered in $[s,t)$.

Assume A_t has stationary increments, *i.e.* for $t\epsilon\Re, s_1 < s_2 \cdots < s_n$ the distribution of traffic in

$$[t + s_1, t + s_2) \cdots [t + s_{n-1}, t - s_n)$$

is independent of t. Further, assume $\mathrm{E}[A_t^2] < \infty$. Let $v(t) = \mathrm{Var}(A_t)$.

Then $\mathrm{Cov}(A_s, A_t) = \frac{1}{2}\left(v(t) + v(s) - v(t-s)\right) \quad , \quad s < t.$

The process A_t is *short-range dependent* if for $s < t \leq u < v$ the correlation coeefficient between $A(\alpha s, \alpha t)$ and $A(\alpha u, \alpha v)$ converges to zero as $\alpha \to \infty$. Otherwise the process is *long-range dependent*.

The Ethernet data gives variance curves $v(t) = t^p$, $1 < t < 2$ over many time scales. Thus,

$$\mathrm{Var}(A_{\alpha t}) = v(\alpha t) = (\alpha t)^p = \alpha^p t^p = \alpha^p \mathrm{Var}(A_t)$$

revealing a similar correlation structure over values of α.

A process Y_t is (strictly) self-similar, with Hurst parameter H ($\frac{1}{2} \leq H < 1$) if, for any $\alpha > 0$, $Y_{\alpha t}$ and $\alpha^H Y_t$ have the same finite-dimensional distributions.

The presence of self-similarity changes the tail probability of a buffer from

$$\mathrm{P}(\text{buffer size} > x) \sim \mathrm{e}^{-\eta x}, \quad x \to \infty, \quad \eta > 0 \tag{2.32}$$

to

$$\mathrm{P}(\text{buffer size} > x) \sim \mathrm{e}^{-\gamma x^{\beta}}, \quad x \to \infty, \quad 0 < \beta < 1 \tag{2.33}$$

where $\beta = 2 - 2H$, assuming a Fractional Brownian motion model [42]. Effective bandwidths relying on the dominant eigenvalue take the form of (2.32). When long-range dependence or self-similarity is present they can *underestimate* cell loss ([14], [5]). In [38] some of these results are summarized, developing an effective bandwidth formula for self-similar or long-range dependent traffic:

$$\hat{C} = R_{mean} + \left(K(H)\sqrt{-2\log(CLR)}\right)^{\frac{1}{H}} \sigma^{\frac{1}{2H}} x^{-\frac{1-H}{H}} R_{mean}^{\frac{1}{2H}} \tag{2.34}$$

where σ is the variance and $K(H) = H^H(1-H)^{(1-H)}$. There has been no result to show that this 'effective bandwidth' is additive as in (2.16).

In [5] VBR video traffic is studied for long-range dependence. The study looked at 20 video sources from a variety of types and lengths. The clear result is the presence of long-range dependence in the VBR video data they investigated. Interestingly, one of the traces studied is the video teleconference trace used in [20]. It seems that the effective bandwidth from [12] based on the DAR(1) model of the trace works well even in the presence of long-range dependence.

Unlike Ethernet data, VBR video traces show strong short-term correlations. Thus, VBR video traffic appears to be only asymptotically self-similar [14]. Over small time-lags the short-term correlations are significant, which is consistent with Markov and DAR models. In the video teleconference data studied in [20] long-range dependence becomes important beyond lags of about 6 seconds. An attempt is made in [29] to bridge the short-range dependent results of [20] and the long-range dependent results of [42] and [14]. Their findings suggest that there is a 'crossover' effect for which beyond a certain number of sources, long-range dependent sources (high H) require less bandwidth than short-range dependent sources (small H), when all other factors are the same. This result was proven for the fractional Brownian motion model (which is *exactly* self-similar) in [30]. The significance of this result is that beyond some threshold short-range dependent (*i.e.* DAR) models, which yield effective bandwidths, are conservative. For traffic engineering this is a safe (albeit wasteful) region. Conversely, on the small side of the threshold long-range dependence will mean higher cell losses. An area of future research is to find this threshold based on known traffic parameters for the sources.

2.4.4 Discussion

Modelling VBR traffic sources for network engineering poses serious challenges. The effects of long-range dependence and the complexity of video sources have proven to be significant and create obstacles to the use of effective bandwidths. Without a simple and accurate traffic model for video sources, the use of effective bandwidths for real-time CAC seems unlikely. Even the techniques in [11] and [12] for DAR(1) models of teleconference traffic would seem to require a lot of computation. An alternative may be to declare the effective bandwidth of the source at contract negotiation. In this context measurement-based CAC methods, like Dynamic CAC, that don't require apriori knowledge of the cell arrival process, look more attractive despite their additional complexity.

Chapter 3

CAC for On-Off Type Sources

3.1 Overview of the Simulations

To compare static and Dynamic CAC algorithms an on-off simulation engine was written in C. This allowed observing multiplexer behavior when an unknown and variable number of sources were active. Such sources might represent voice, data, or some future unknown service, depending on the parameters used. Call arrivals were Poisson and call holding times were exponentially distributed. The mean for both of these were allowed parameters. Established calls alternated between burst and idle states. It was assumed that the mean burst length was a declared traffic descriptor to permit calculation of the effective bandwidths.

The program iterated on the elapsed time, and the time step was chosen to be the time required to transmit one cell on the link. For a 150 Mbps link this gives:

$$t_{step} = \frac{424}{150 \times 10^6}$$

In other words, each iteration removed one cell from the non-empty buffer. Henceforth this time period will be called a *slot*. This time step is appropriate if the speed of the link is much large than the peak rate of the simulated sources.

In the standard on-off flow approximation each source produces a flow of bits. For the simulation this flow was 'packetized' in the sense that each source produced cells in the burst state every $\frac{424}{R_{peak}}$ seconds. This was necessary because Dynamic CAC is cell-based, and seemed natural for simulating an ATM environment. Issues relating to this packetization are discussed below.

At the start of the simulation many sources can be accepted, and their burst and idle states coincide somewhat closely, as determined by their exponential distributions. This leads to non-stationary behavior in which a large number of sources are in the burst state simultaneously. While this unnatural situation is well-handled by peak-rate based CAC methods, it is problematic when a stationary arrival process is assumed. To observe stationary behaviour the program was run approximately 70 seconds without recording statistics.

Statistics were recorded every 35377 slots, which is about 100 ms for a 150 Mbps link. Observed statistics included the current and maximum number of sources, call blocking probability, cell loss probability, link utilization and instantaneous buffer utilization. Link utilization was calculated as $\frac{\text{Total Cells - Cells Lost}}{\text{Number of Slots}}$. Instantaneous buffer utilization was calculated as $\frac{\text{Current Buffer Size}}{\text{Maximum Buffer Size}}$. These statistics were recorded in a format suitable for plotting. Other summary statistics were recorded at the end of the simulation to provide additional information and a check of consistency. This information included the number of slots, total number of cells and cells lost, and the number of calls accepted and rejected.

Although any parameters could have been specified for the On-Off sources, parameters from [1] were typically used. The source has peak rate 4 Mbps, utilization 0.2, and mean burst length 100 ms. This provided a reasonable number of sources, keeping execution times reasonable.

The on-off source engine became the foundation around which different CAC methods were added. For example calculating effective bandwidths is straightforward while Dynamic CAC requires more statistics and distributions to be recorded.

Many of the simulations were run on a Pentium-Pro computer under Linux. These were primarily the simulations of Dynamic CAC requiring seven executions. When the chosen parameters meant the simulation was computationally intensive an Ultra Sparc 1 computer was used. Other simulations were performed on Sparc 10, 5 and 4 computers, with faster computers chosen for more intensive simulations. Most of the simulations ran overnight, from 6 to 18 hours, depending on the chosen parameters and the computer's load.

3.2 Confirmation of the Program

To confirm that the on-off simulation engine was producing reasonable results a test was performed. Effective bandwidths are based on the assumption of an infinite buffer (Section 2.2.2). For a single source the actual probability Pr(buffer content > x) is known to be close to the overflow probability ε when x and ε are used in the effective bandwidth calculation. For example, if the desired overflow is $\varepsilon = 10^{-6}$ for a buffer size of $x = 3 \times 10^6$ bits and the link capacity is taken to be the effective bandwidth for a chosen source, then Pr(buffer content $> 3 \times 10^6) \approx 10^{-6}$. Moreover, a plot of x vs. log(Pr(buffer content $> x$)) should show a linear relationship with negative slope.

With this in mind the on-off engine was modified to allow an infinite buffer, one source with infinite holding time, buffer size of 3×10^6 bits and desired overflow 10^{-6}. The effective bandwidth was calculated to be 22,250,531 bps. This rate was used as the link capacity. The program simulated 70,000 sec with a slot length of 0.00001 sec. For comparison the program was also modified to work with a pure bit flow with the same link capacity. The flow version simulated 70,000 sec and 10^7 sec with a slot length of 0.001 sec. The results (Figure 3.1) show a definite linear relationship, and fall-off with short simulation times. This is consistent with the low tail probabilities in the exponential distribution used for the burst lengths. Results for the cell-based simulation show smaller buffer overflow probabilities. Equivalently, the calculated effective bandwidth appears to be a bit more than is necessary for the desired overflow. I take this to be the effect of packetization which adds or removes blocks of 424 bits (one cell) at a time.

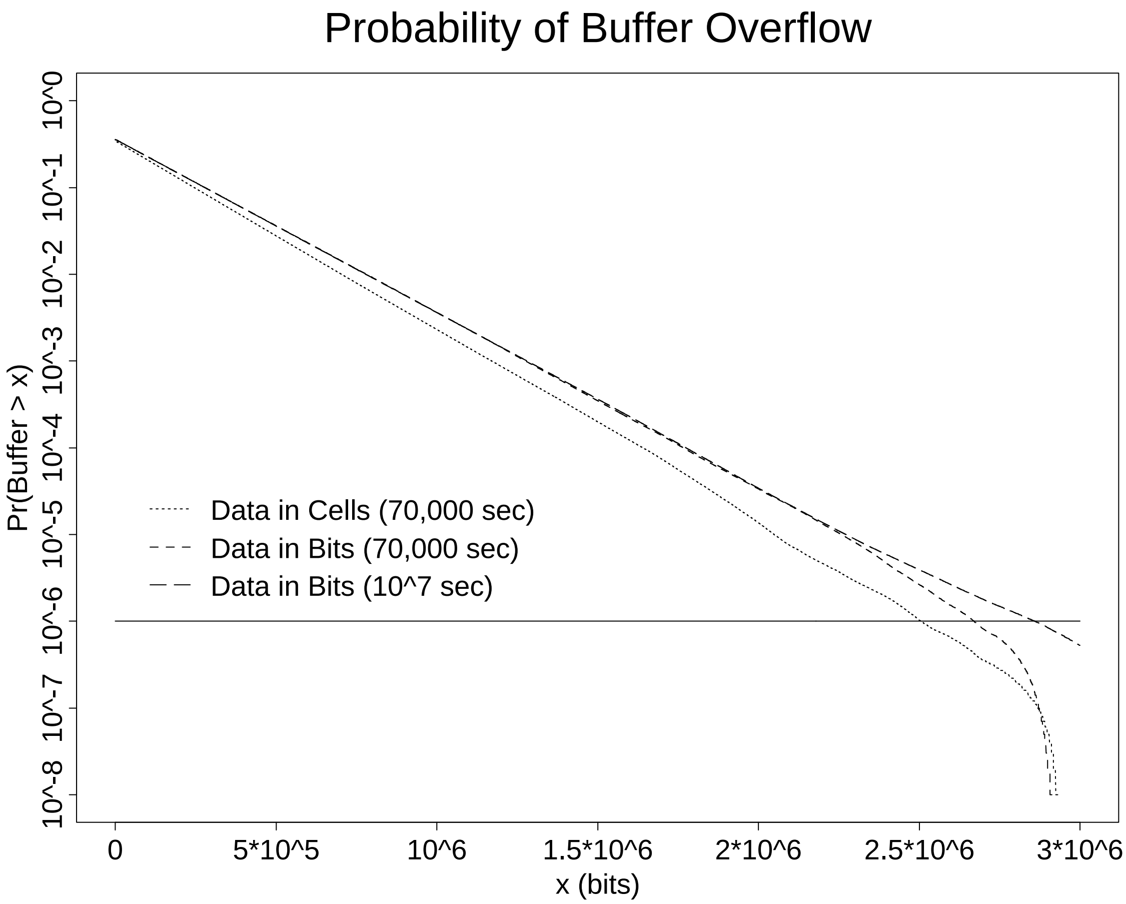

Figure 3.1: Buffer Overflow in Cell-Based and Flow-Based On-off Engines

3.3 Performance of Static CAC

It was decided to first compare CAC methods using with a buffer size of 7075 cells (approximately 3 Mbits) and desired overflow 10^{-6}. This is similar to an arrangement discussed in [17], except desired overflow of 10^{-5} was used. The link capacity is 150 Mbps throughout. It was found by simulation that a fixed number of sources between 150 and 160 could be supported (Figure 3.2).

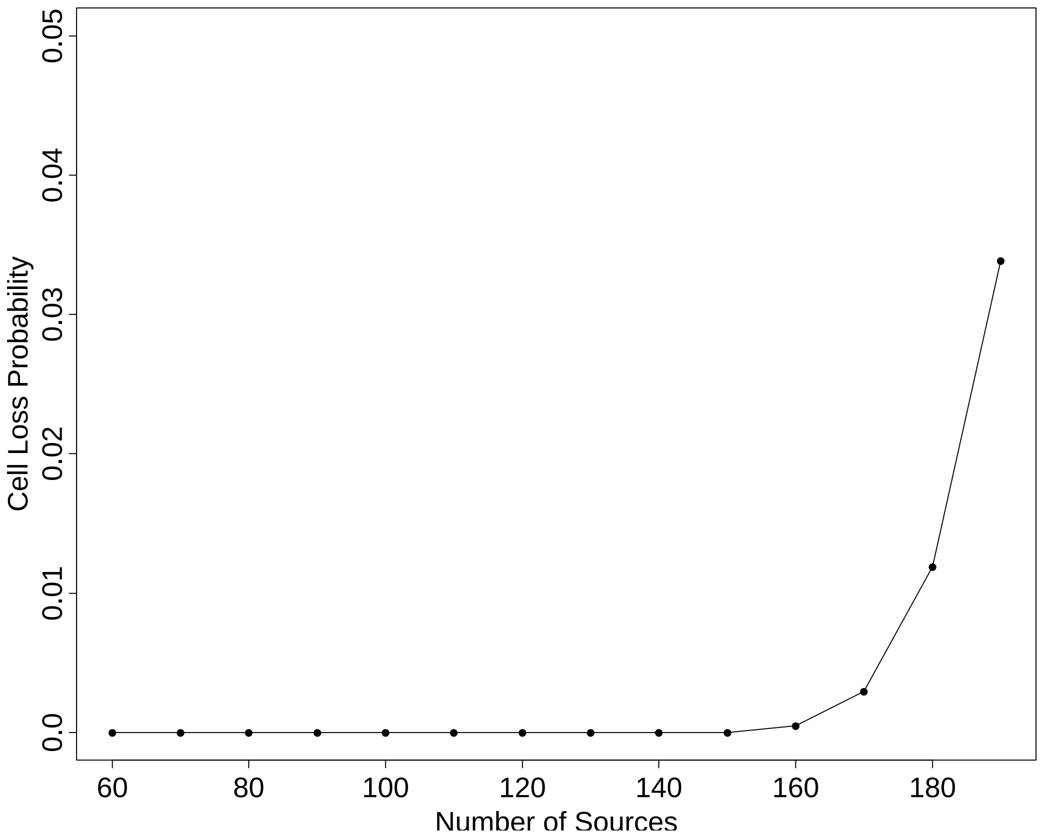

Figure 3.2: Cell Loss Probability for fixed number of On-Off sources, 7075 cell buffer

The required capacity needed for different numbers of sources using peak-rate based CAC as well as the on-off flow approximation or the pure Gaussian approximation was calculated (Figure 3.3). Ordinarily the minimum of the flow and Gaussian approximations would be used. For comparison the required bandwidth based on average rates is also shown. At 150 Mbps, 37 calls by peak-rate, 66 calls by effective bandwidths, and 90 calls by effective bandwidths plus Gaussian approximation can be supported. As a matter of interest, simplified logarithm CAC would allow 71 sources, while logarithm CAC would allow 110 sources.

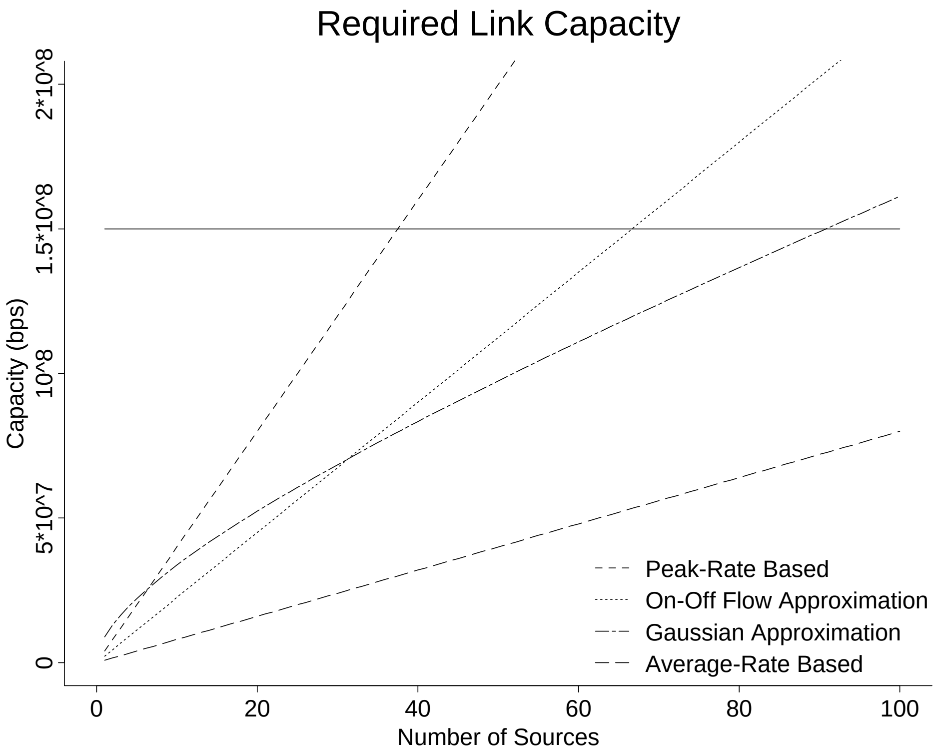

Figure 3.3: Required capacity for On-Off sources, 7075 cell buffer and 10^{-6} overflow

In simulation, calls arrived with mean intensity 100 per second, and mean holding time 3 sec. Seven executions of the program were performed for each of the three CAC methods (Figure 3.4 – Figure 3.6). Statistics were recorded for the period 70 sec to 170 sec. All three methods are highly conservative as can be seen from the consistently low buffer utilization. Cell loss is not a problem. From the link utilization and blocking probability the effective bandwidth plus Gaussian approximation CAC method is best, while peak-rate based CAC is the worst. This is not suprising in view of the maximum number of simultaneous calls previously calculated for each method.

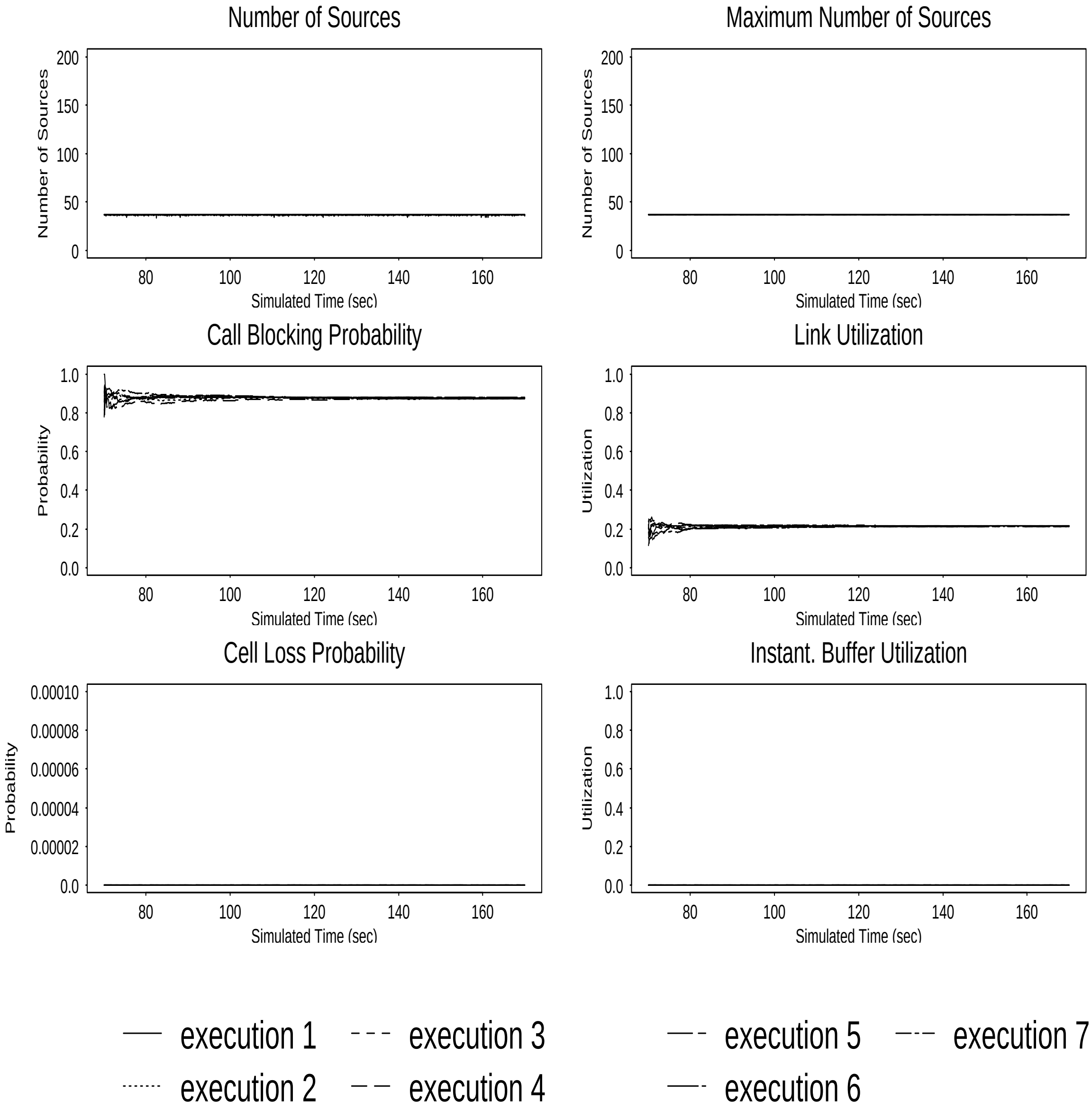

Figure 3.4: Peak-rate Based CAC for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

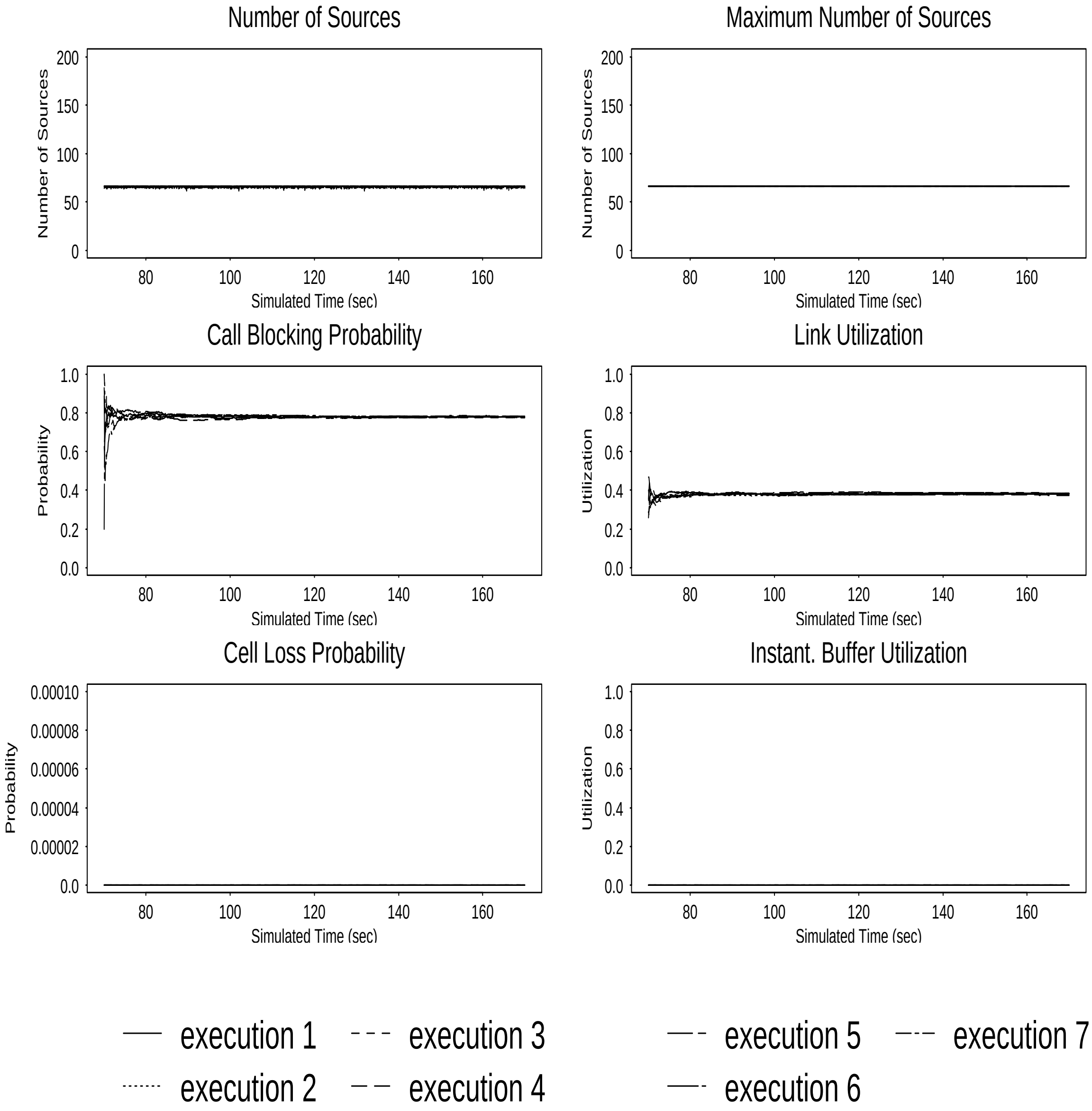

Figure 3.5: Effective bandwidth CAC for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

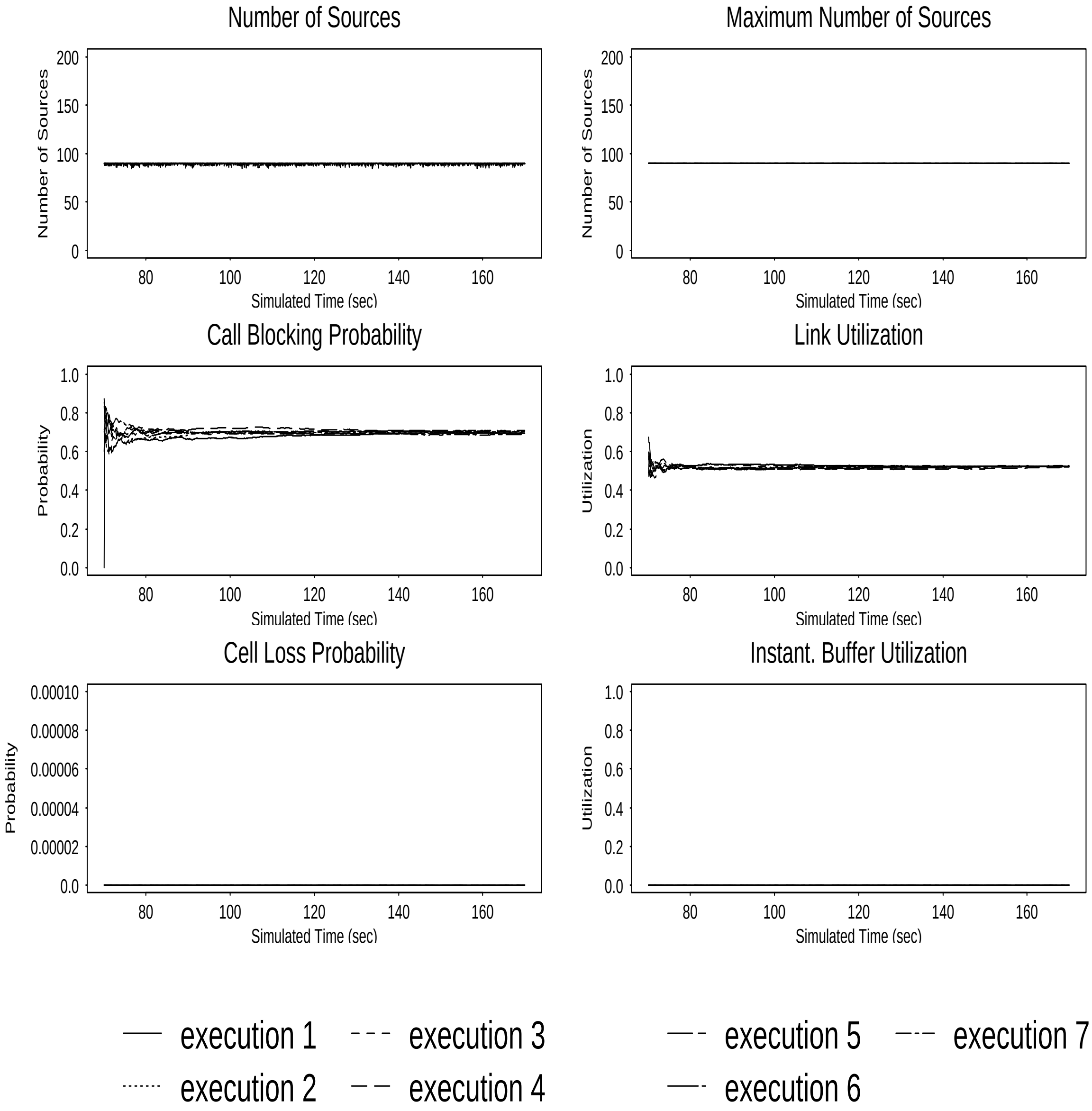

Figure 3.6: Effective Bandwidth plus Gaussian Approximation CAC for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

3.4 Dynamic CAC

To compare Dynamic CAC with the static CAC results shown previously it was necessary to find suitable values for the parameters α, r, and M. A result not anticipated at the outset of this work was the sensitivity to choices of these parameters. Some parameter combinations give cell loss much higher than the level desired. It is useful to recall that Dynamic CAC uses a bound on cell loss that replaces the exact probability distribution with its estimate. A poor estimate has the potential for causing unacceptably high cell loss.

As with the static CAC simulations, a buffer size of 7075 cells and desired cell loss 10^{-6} was used. Calls had a mean holding time of 3 seconds and a mean call arrival rate of 100 per second. The relatively high call arrival rate was chosen to test the effectiveness of the CAC method. There is an element of opportunism to measurement-based CAC methods. If too many calls are admitted in times of low activity then cell loss may occur in times of higher activity. Higher call arrival rates are more likely to demonstrate this weakness.

3.4.1 Sensitivity to r

To see the effect of r, values in [47] were initially used ($\alpha = 0.1$ and $M = 1$. Seven executions were performed for $r = 2, 10, 50, 100, 500, 1000$ and 2000 (Figure 3.7 – Figure 3.13). The performance of Dynamic CAC depends strongly on this parameter, with this particular type of source. When r is too small or too big calls are needlessly blocked, and the link utilization is kept low. This is most acute for small r. For middle range values (r=50, 100) too many sources are accepted. While link utilization is high, it is at the expense of cell loss, which is high in some executions, despite the relationship between r and buffer size described in [47] and [48] (section 2.3.1).

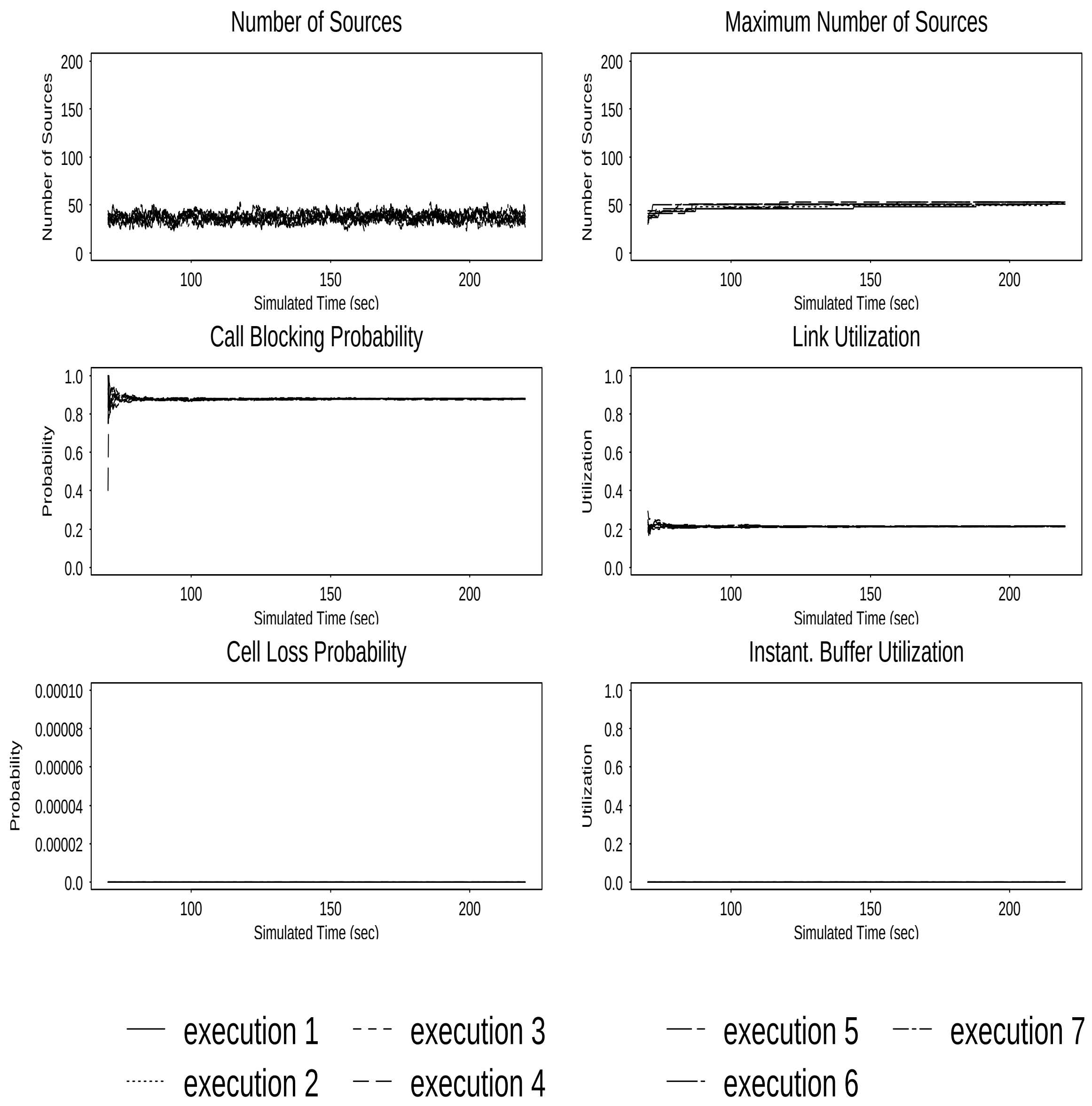

Figure 3.7: Dynamic CAC ($\alpha = 0.1, M = 1, r = 2$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

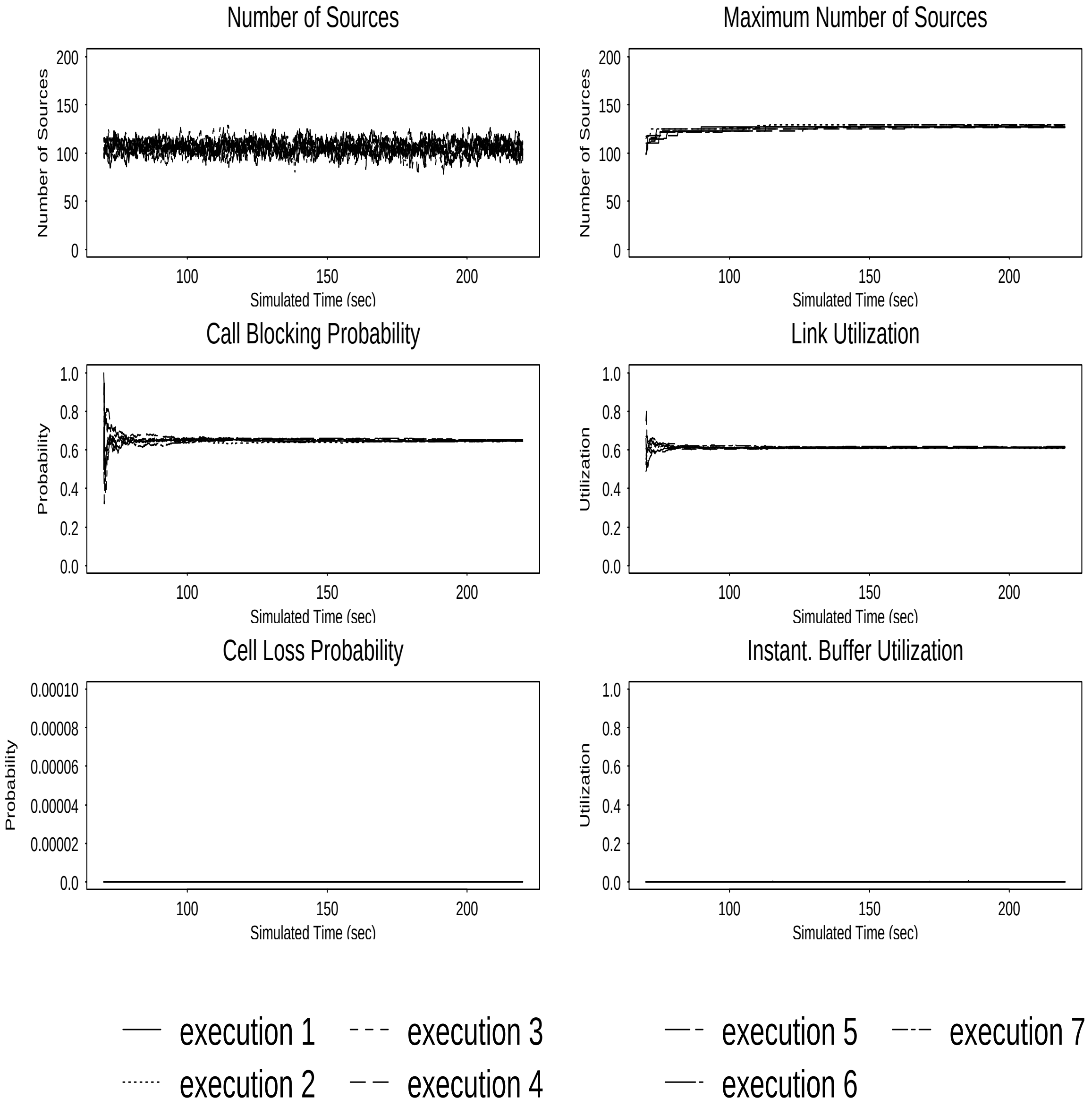

Figure 3.8: Dynamic CAC ($\alpha = 0.1, M = 1, r = 10$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

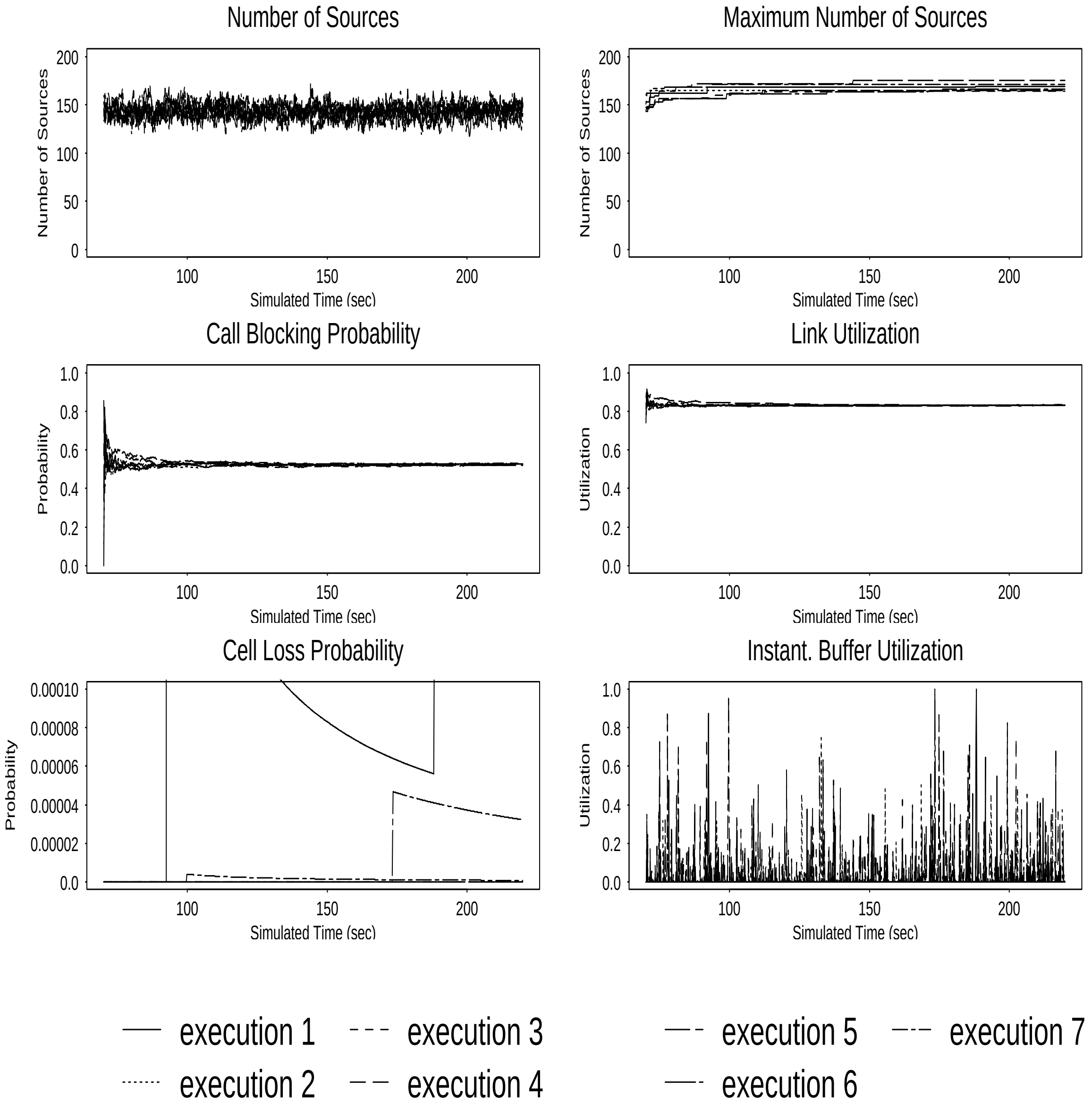

Figure 3.9: Dynamic CAC ($\alpha = 0.1, M = 1, r = 50$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

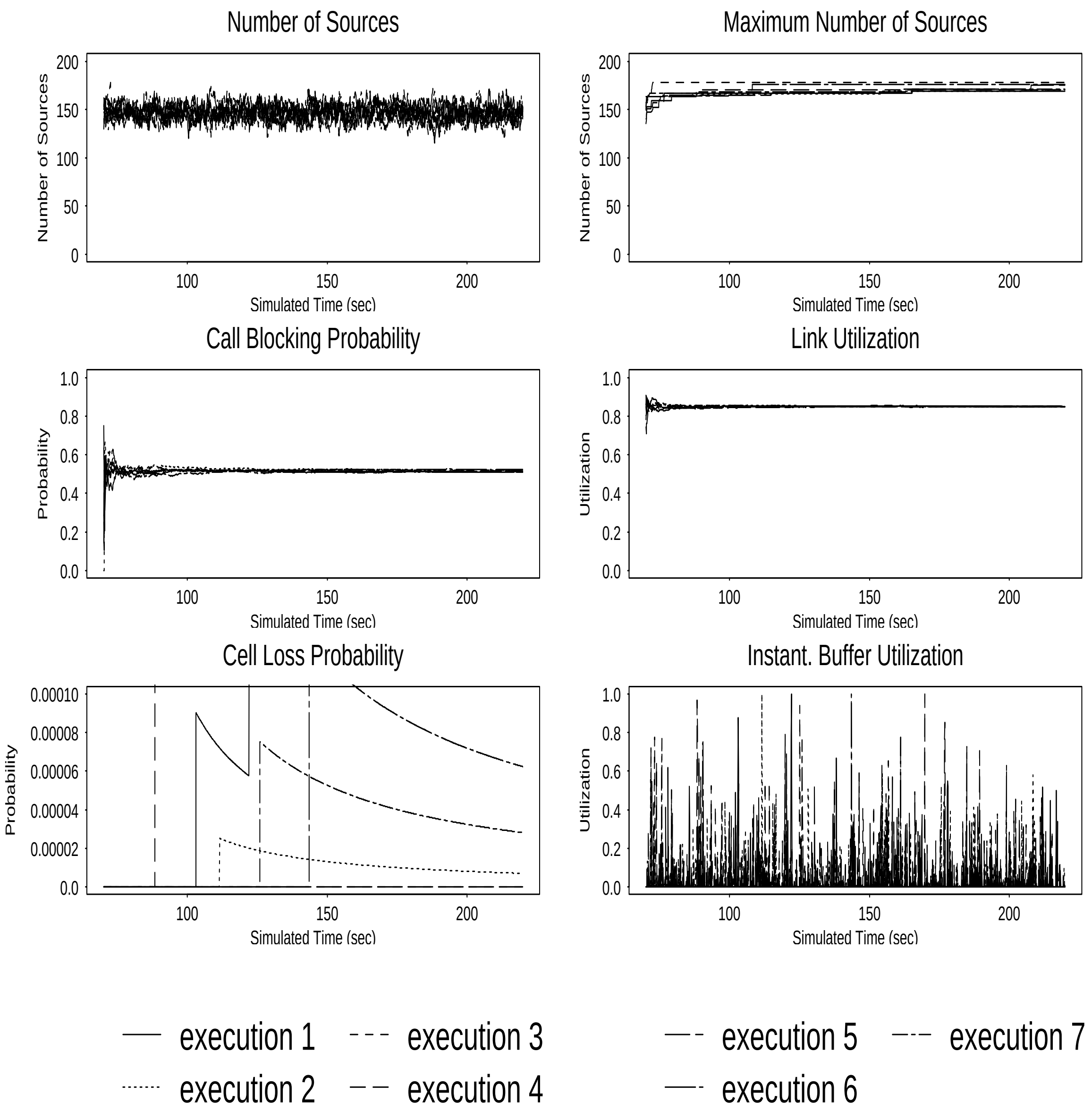

Figure 3.10: Dynamic CAC ($\alpha = 0.1, M = 1, r = 100$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

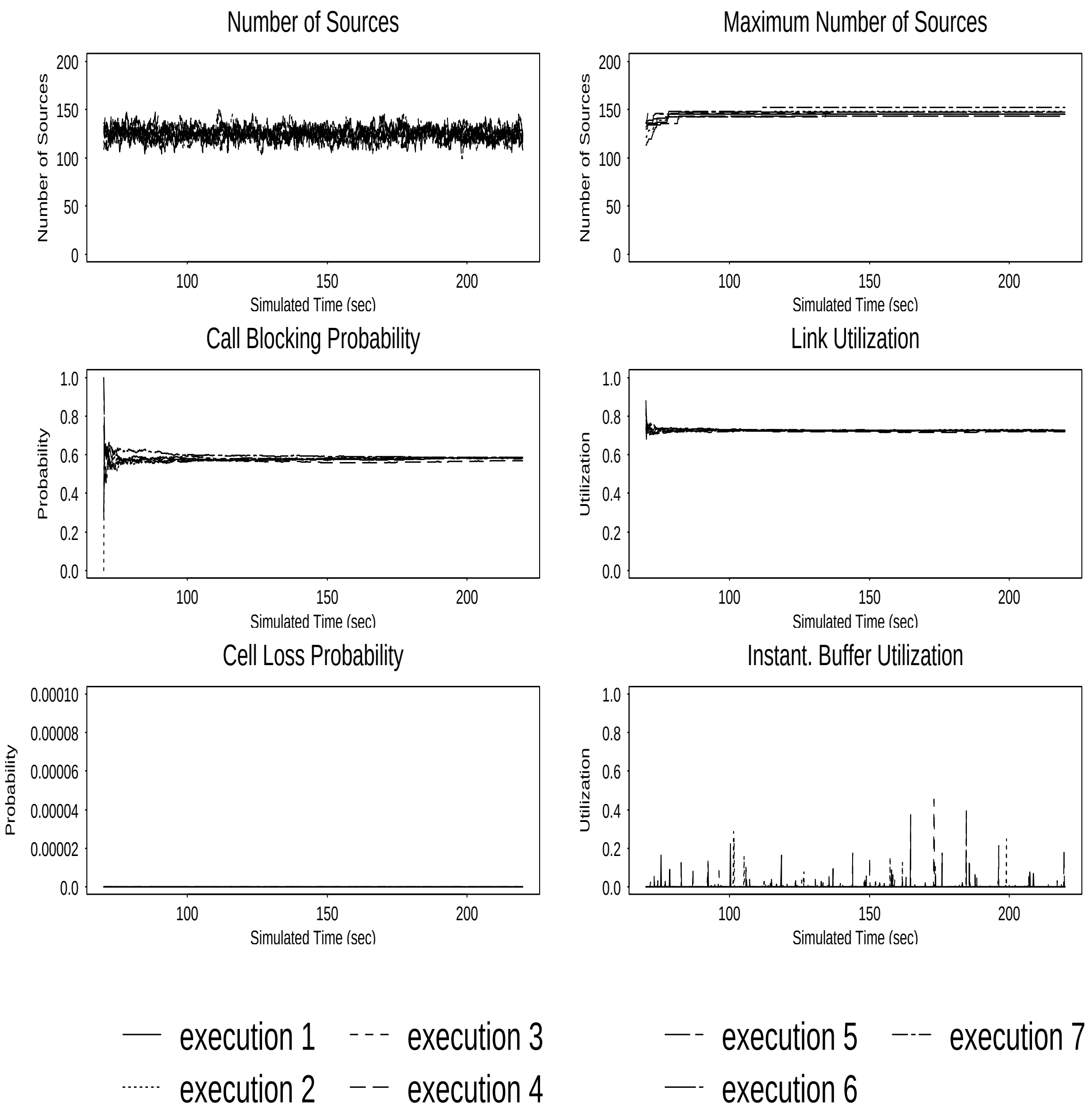

Figure 3.11: Dynamic CAC ($\alpha = 0.1, M = 1, r = 500$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

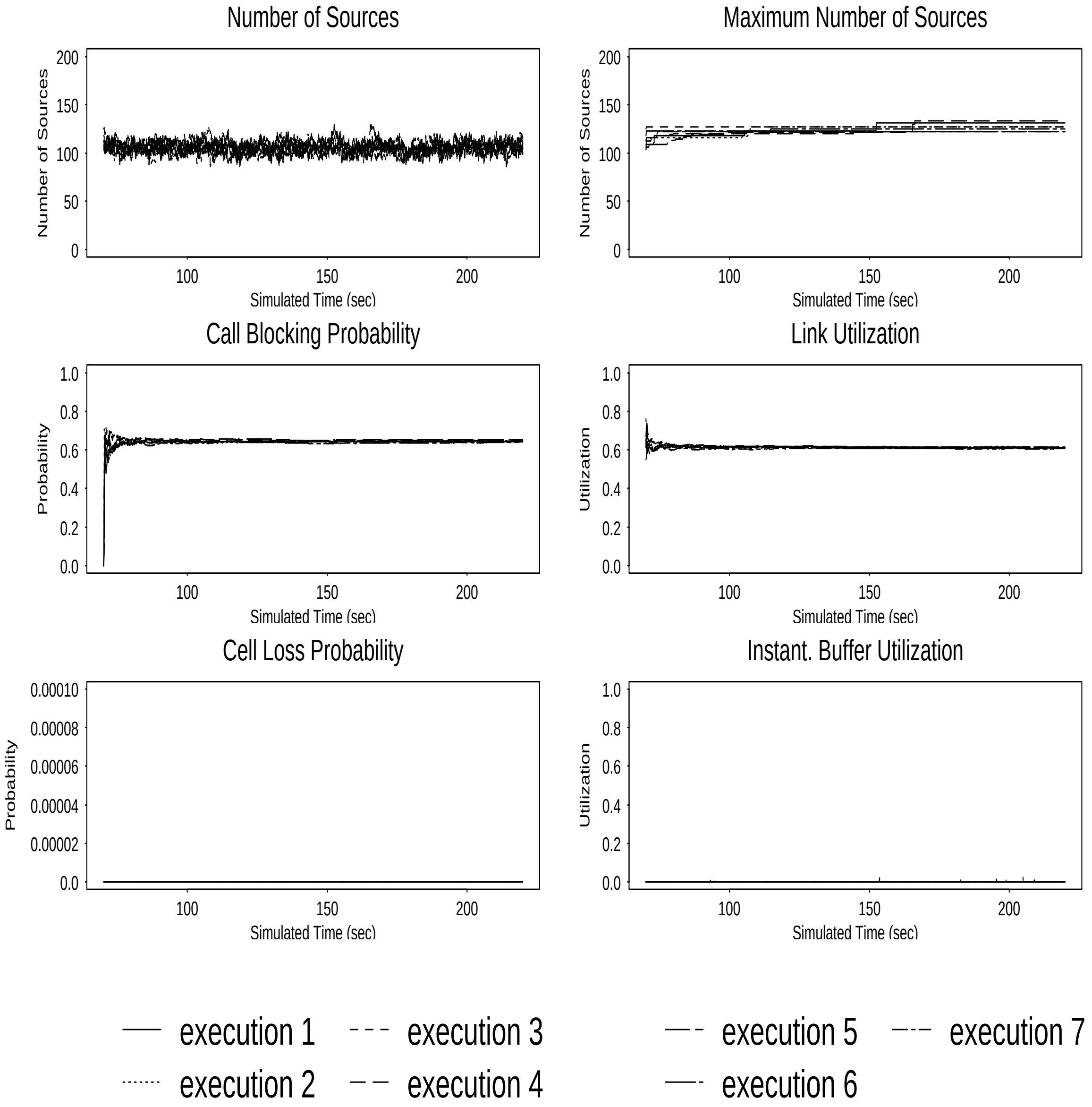

Figure 3.12: Dynamic CAC ($\alpha = 0.1, M = 1, r = 1000$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

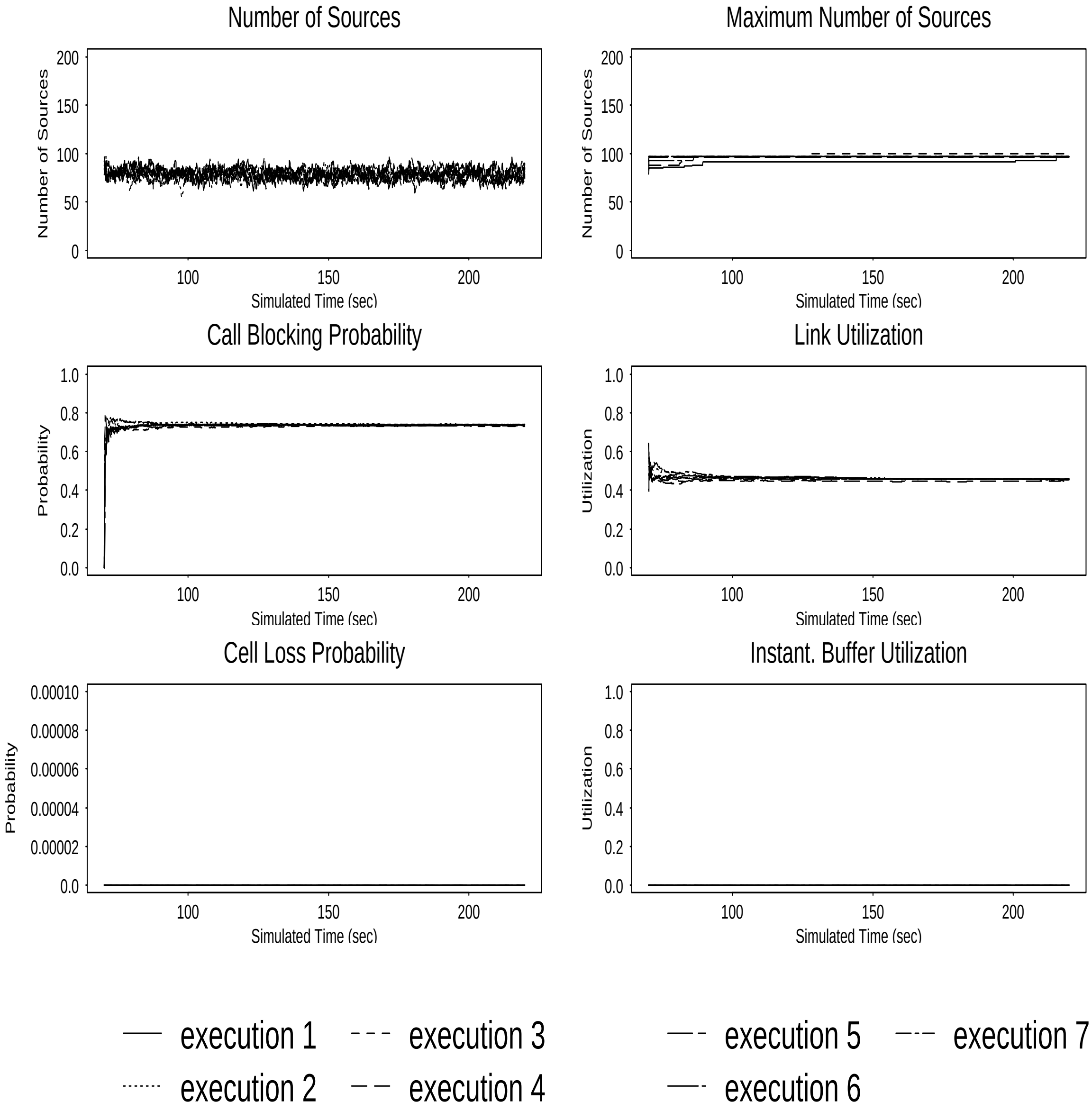

Figure 3.13: Dynamic CAC ($\alpha = 0.1, M = 1, r = 2000$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability

3.4.2 Sensitivity to α and Call Length

To investigate the role of α in Dynamic CAC, one execution was performed for each of the values $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$ (Figure 3.14). As before, calls arrived with mean intensity 100 per second, and their holding times were exponentially distributed with mean 3 seconds. Other parameters were $M = 1$, $r = 500$ with a buffer size of 7075 cells. Values larger than 0.1 admitted too many sources, leading to high cell loss. Increasing α weights the most recent measurement higher, and would be more susceptible to over-admission in times of low activity. This may explain why cell loss increases with increasing α.

Small values of α were tested the same way. Four values were chosen: $\alpha = 0.1, 0.01, 0.001, 0.0001$. While a mean holding time of 3 seconds was initially used (Figure 3.16), other values (1 sec, 10 sec, 30 sec) were also chosen (Figure 3.15, Figures 3.17 – 3.18). Three main observations emerge. The first is that for $\alpha = 0.1$ cell loss becomes a problem for longer call holding times. The second observation is that small values of α (*i.e.* high persistence of old information) leads to fewer admitted sources and lower utilization. The third obervation is revealed by $\alpha = 0.001$. For short holding times (1 sec, 3 sec) the number of sources shows very regularly spaced peaks separated by periods of no sources. The duration of these peaks is 19.7 ± 0.2 seconds. When the holding time is 10 seconds the number of sources shows remnants of these peaks, but they are less defined and less regularly spaced. A likely explanation relates the length of the calls to the persistence of information regarding those calls. Indeed, for short holding times calls may finish but information from their admission based on their peak rates will persist. As the mean holding time increases call lengths become longer. When they are sufficiently large on average, most of the calls will exist beyond the time that the peak rate shift is significant. Even longer persistence for $\alpha = 0.0001$ would explain why calls are admitted initially, and then never again. Future work with heterogenous sources may reveal this to be a significant issue for achieving high utilization in the 'peak-rate only' version of Dynamic CAC. In [46] the declared mean rate is also used. Knowledge of the average rate allows explicit adjustment of the cell arrival distribution when a call is released. Although this technique cannot account for 100% of the cell arrivals due to the released call, it would mitigate the persistence of information from them. Another area for future work is alternatives for updating the cell arrival distribution (equation 2.24).

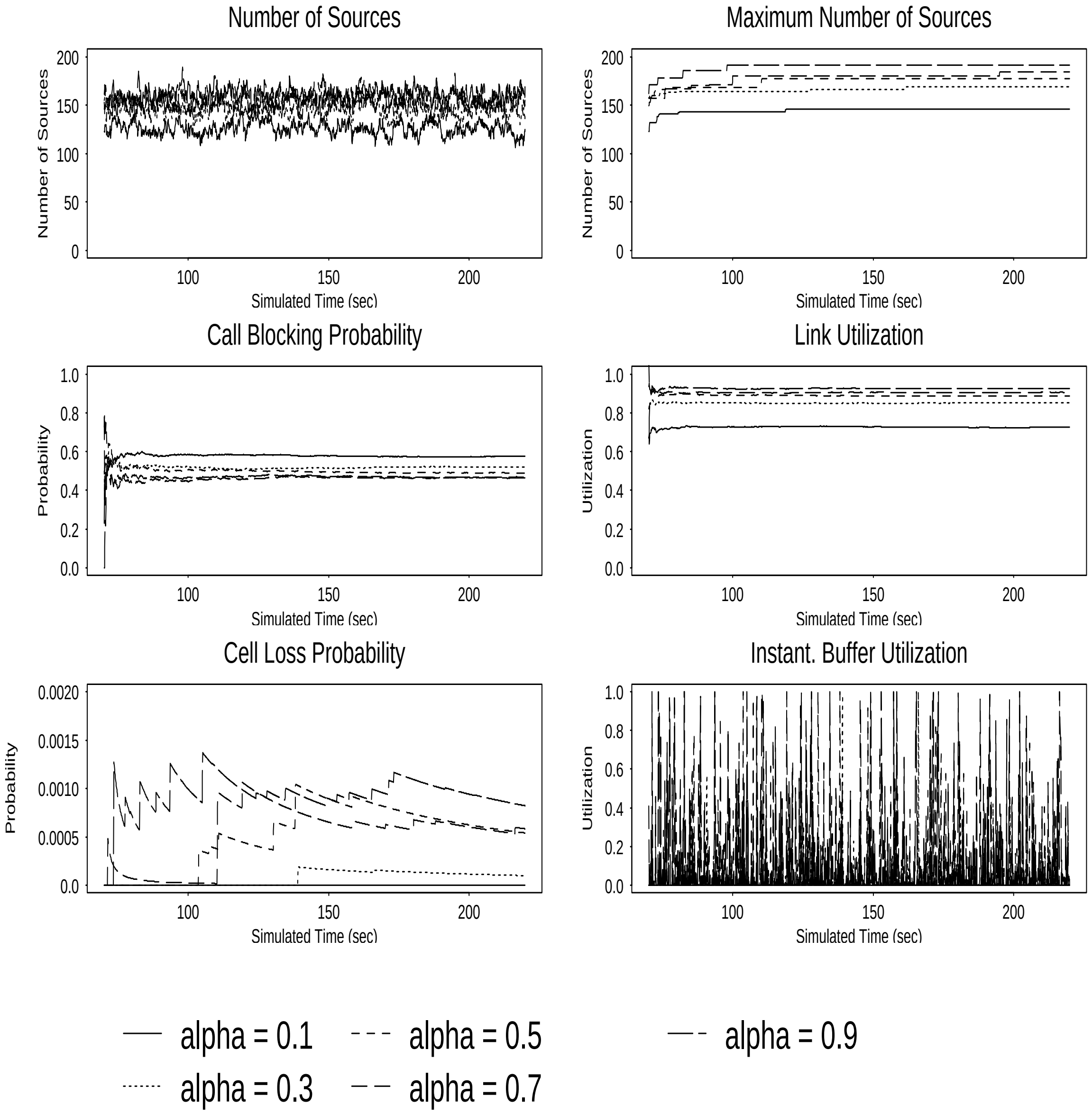

Figure 3.14: Dynamic CAC ($M = 1, r = 500$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds

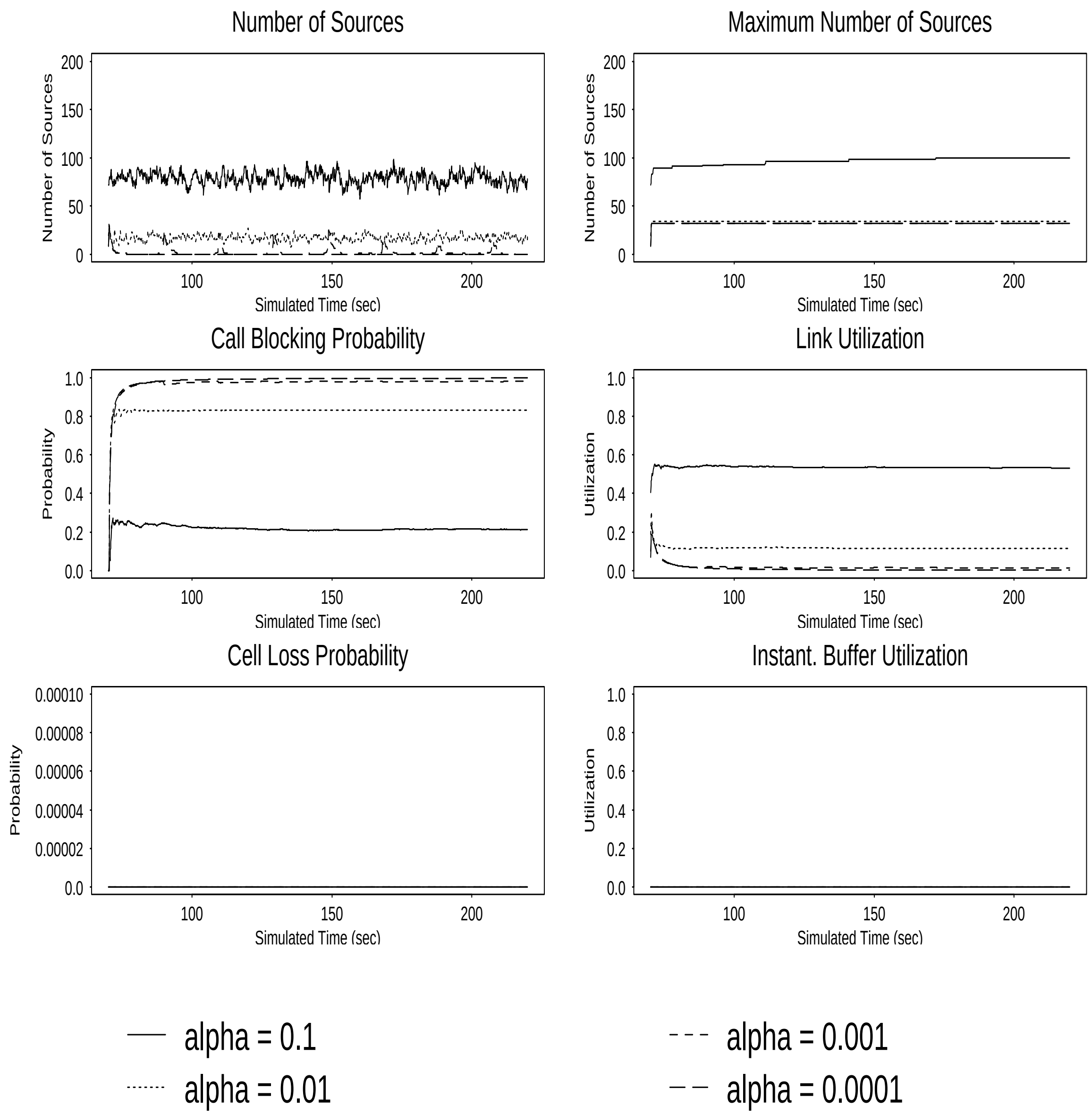

Figure 3.15: Dynamic CAC ($M = 1, r = 500$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 1 second

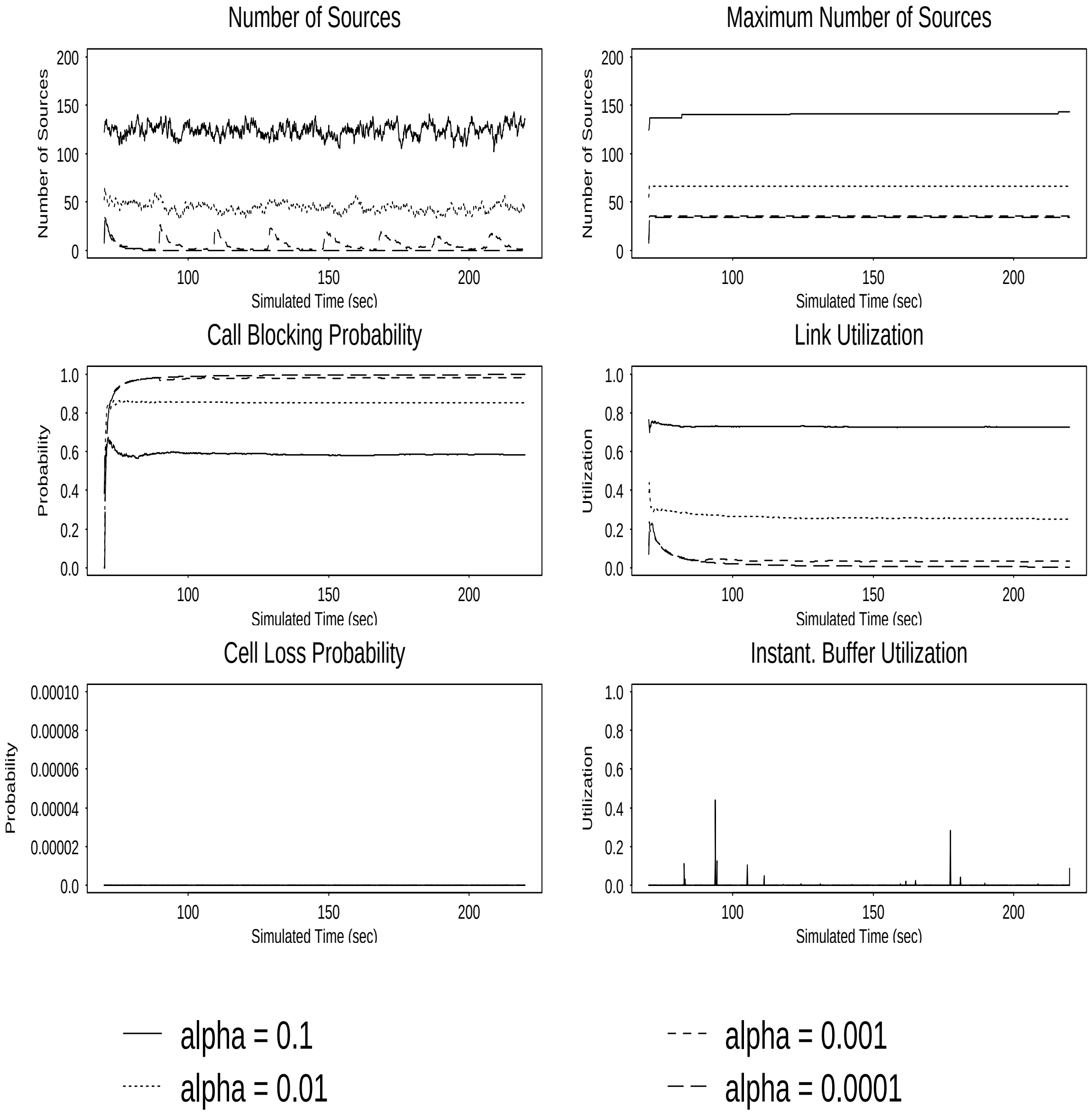

Figure 3.16: Dynamic CAC ($M = 1, r = 500$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds

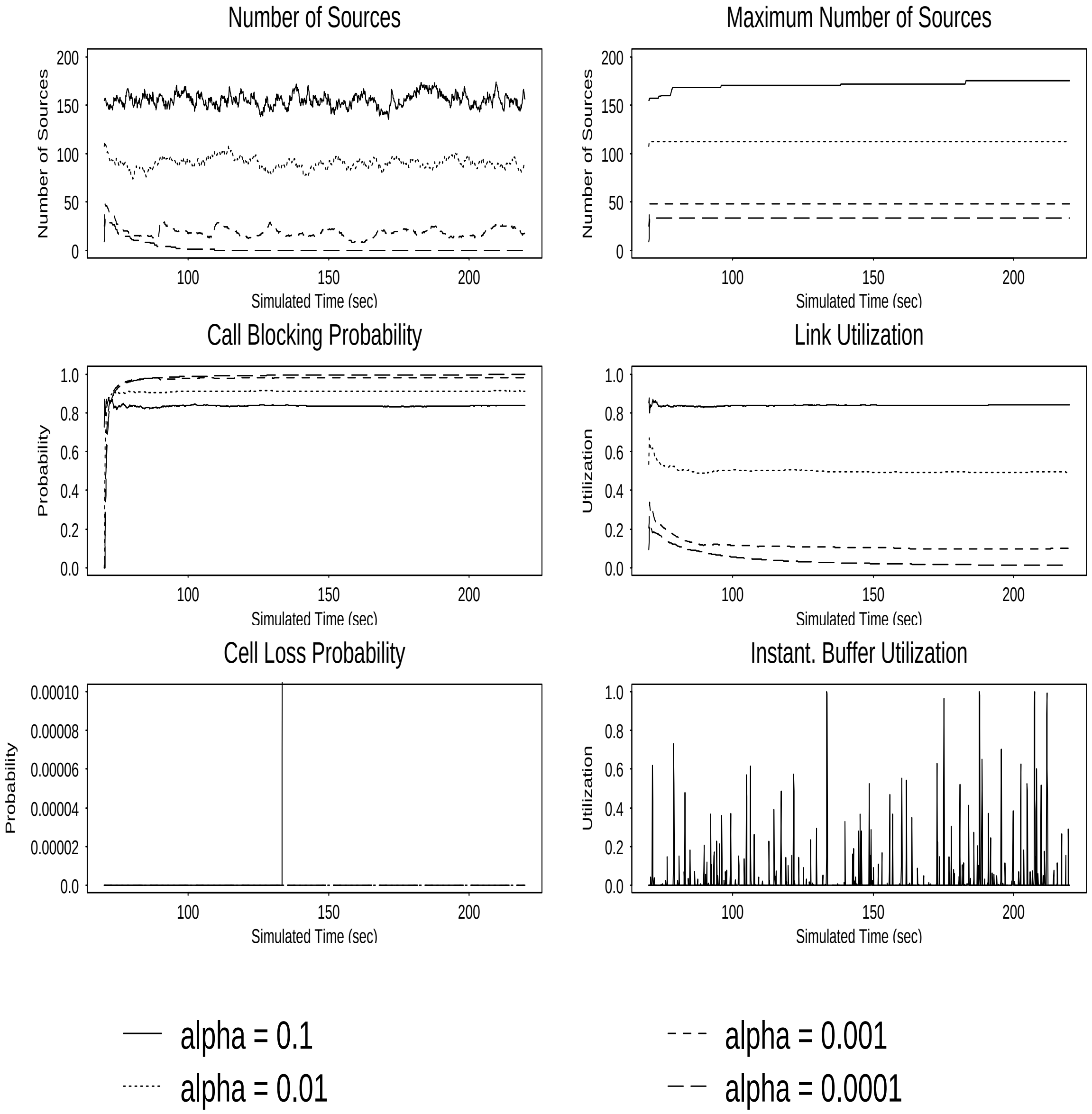

Figure 3.17: Dynamic CAC ($M = 1, r = 500$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 10 seconds

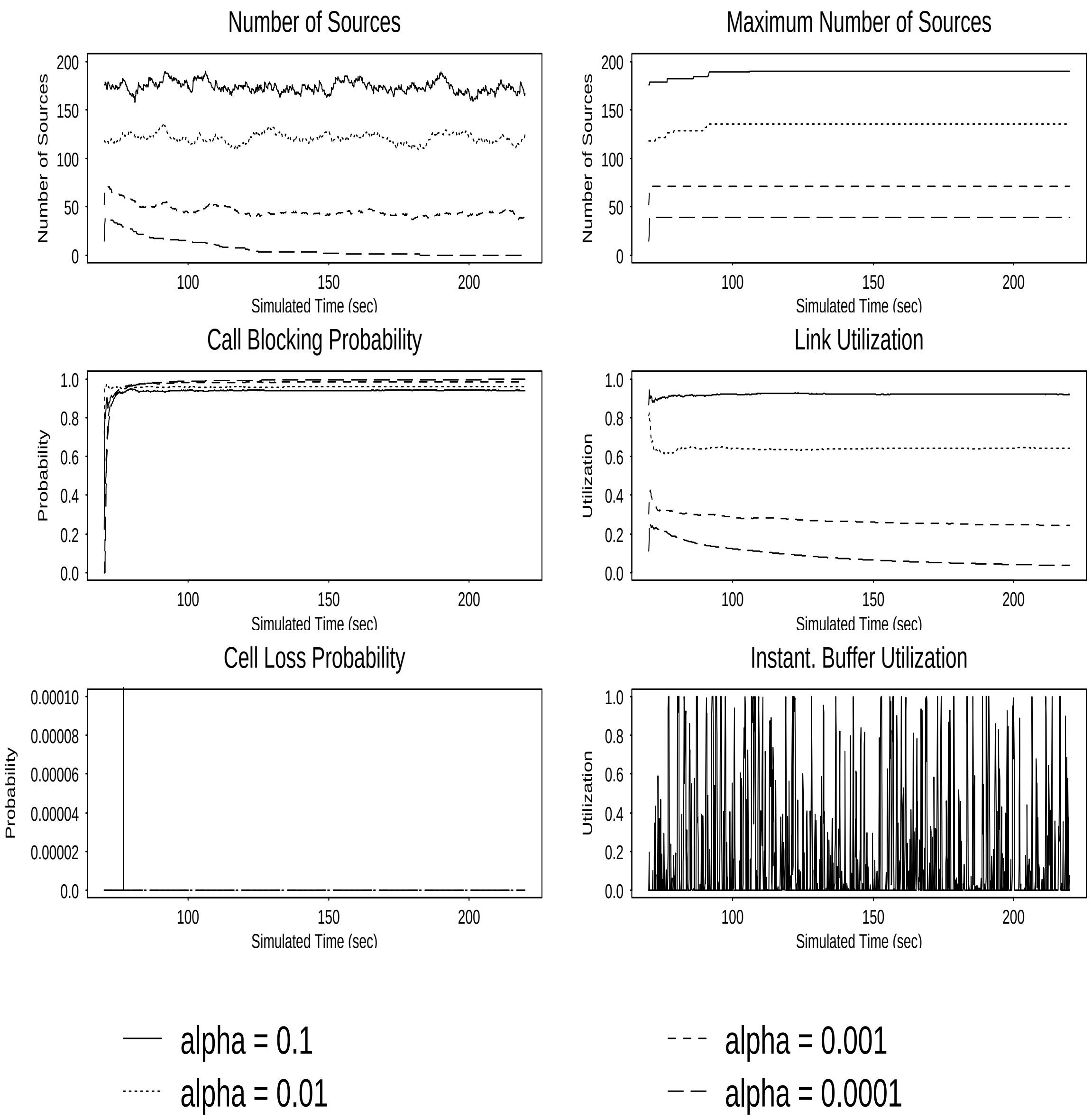

Figure 3.18: Dynamic CAC ($M = 1, r = 500$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 30 seconds

3.4.3 Sensitivity to M

In Dynamic CAC the cell arrival distribution is revised every Mr slots. To investigate sensitivity to M values were chosen such that $Mr = 500$. One of these combinations ($M = 1$, $r = 500$) was presented above (Figure 3.11), and is presented again (Figure 3.19) to simplify comparison. Other values were $M = 5, 25, 50, 100, 500$ (Figures 3.20 – 3.24). Increasing M divides the 500 slot period into more segments for which a count of the number of arriving cells is made. For these simulations $\alpha = 0.1$ was used, and all other values are as previously described.

It can be seen that as M increases (and r decreases) the number of admitted sources decreases to almost zero. When r varied alone the number of sources and link utilization took an optimal value as r increased. Here, when M and r vary together the effect on the number of sources and link utilization is monotonic. Moreover, for $r = 100$ the performance is better when $M = 5$ compared to $M = 1$ (Figure 3.10). Now, cell loss is not a problem. The same is true for $r = 50$. More work is needed to understand the impact of M on the operation of Dynamic CAC.

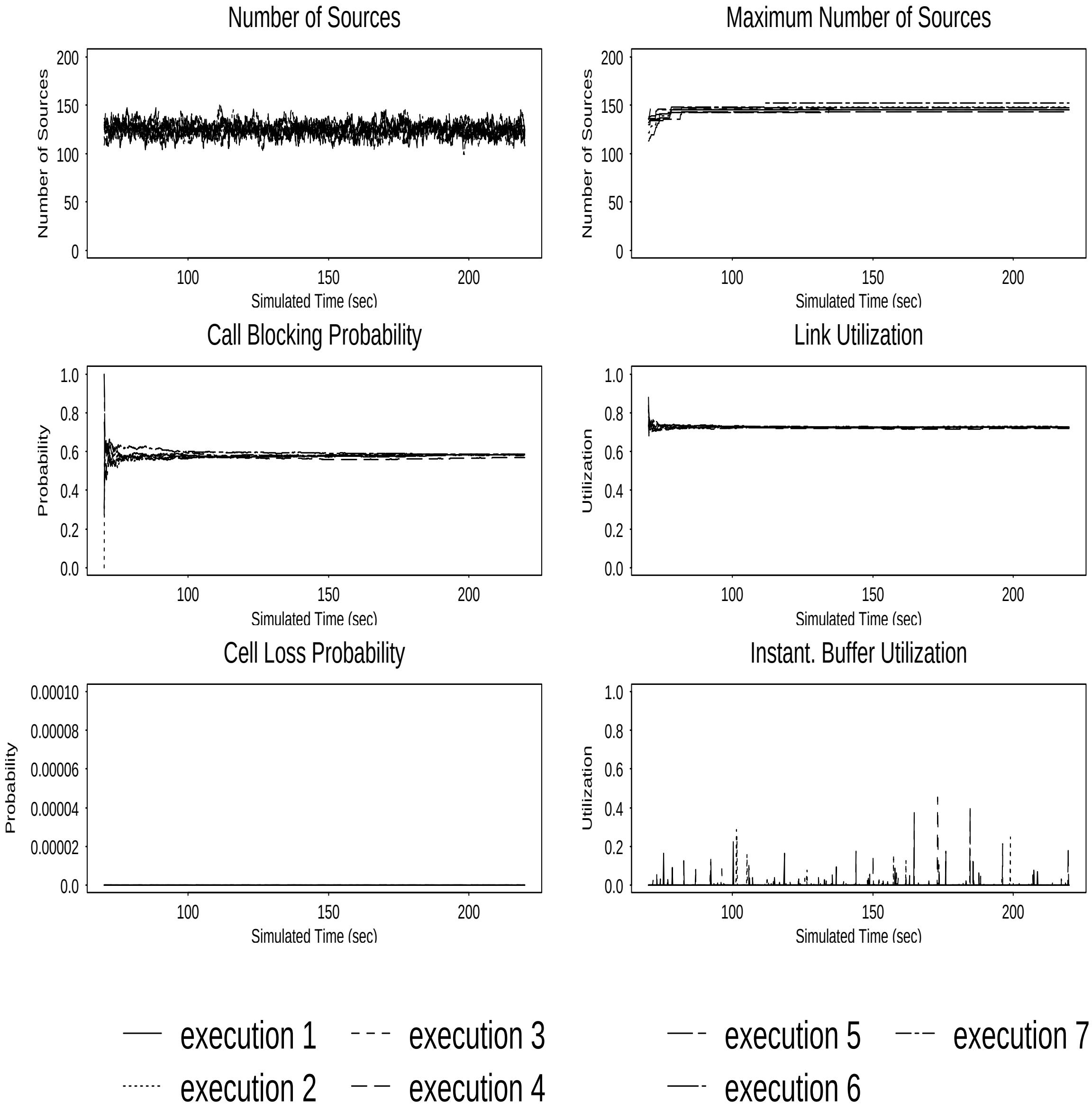

Figure 3.19: Dynamic CAC ($M = 1, r = 500$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds

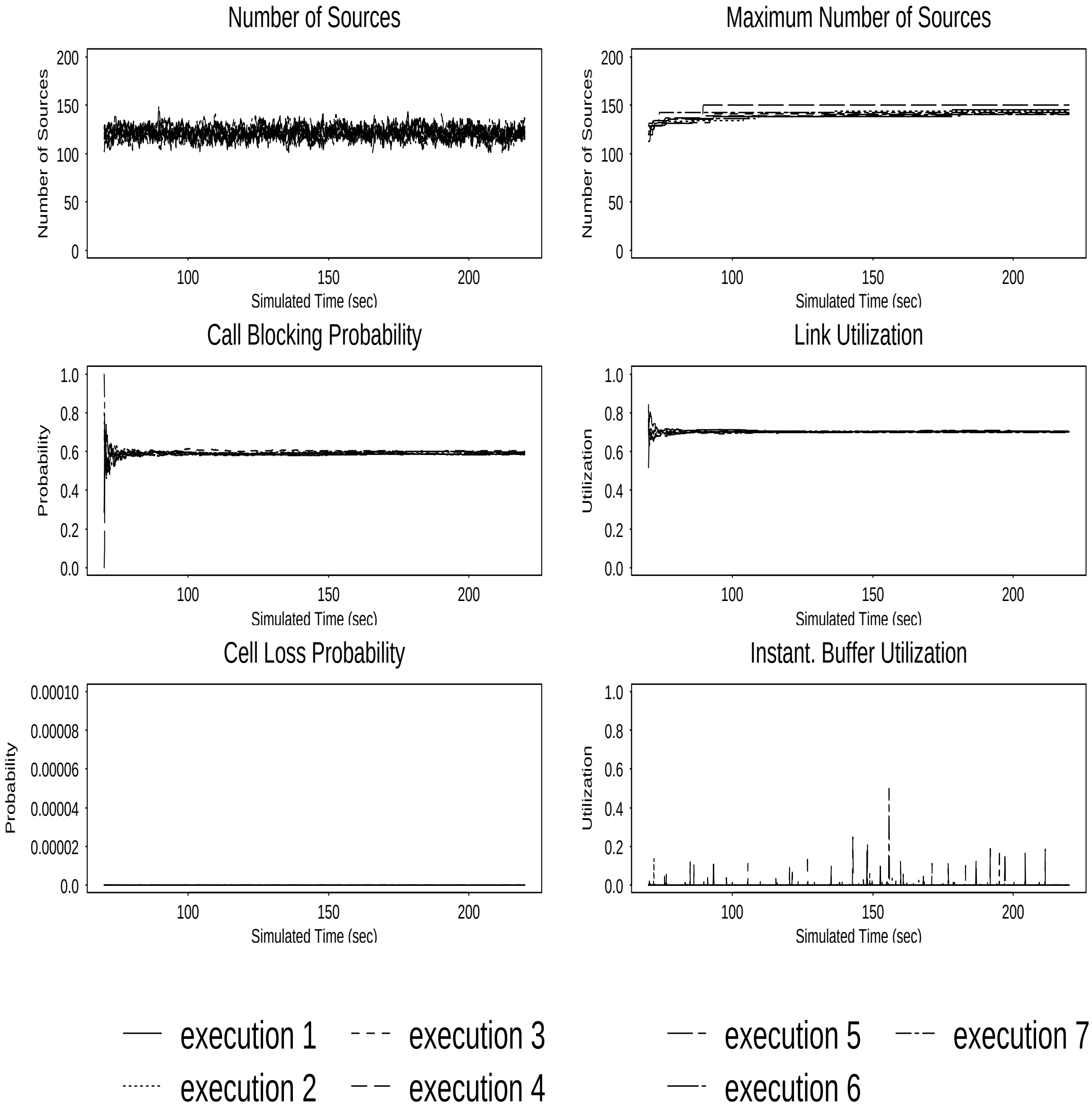

Figure 3.20: Dynamic CAC ($M = 5, r = 100$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds

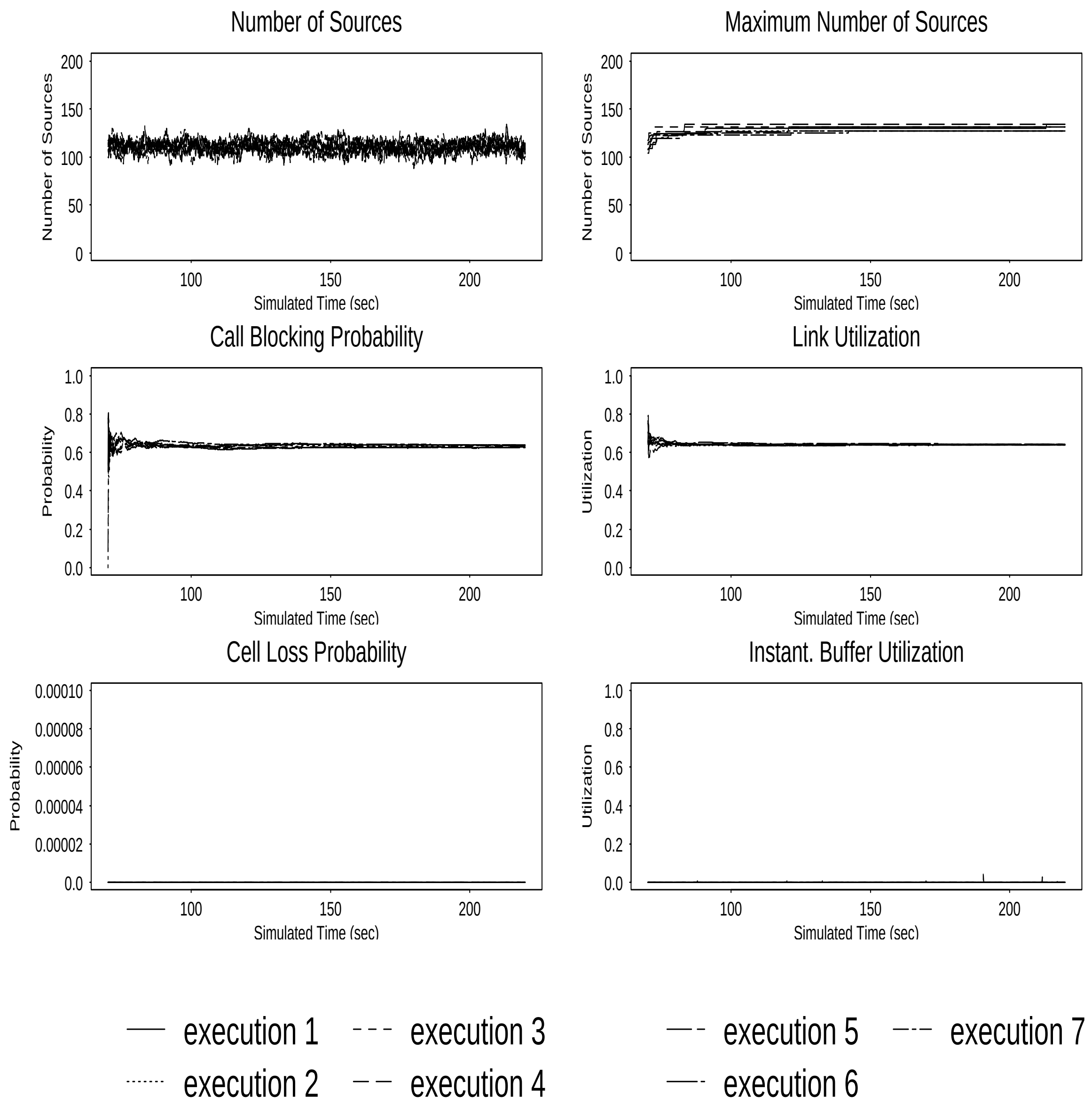

Figure 3.21: Dynamic CAC ($M = 10, r = 50$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds

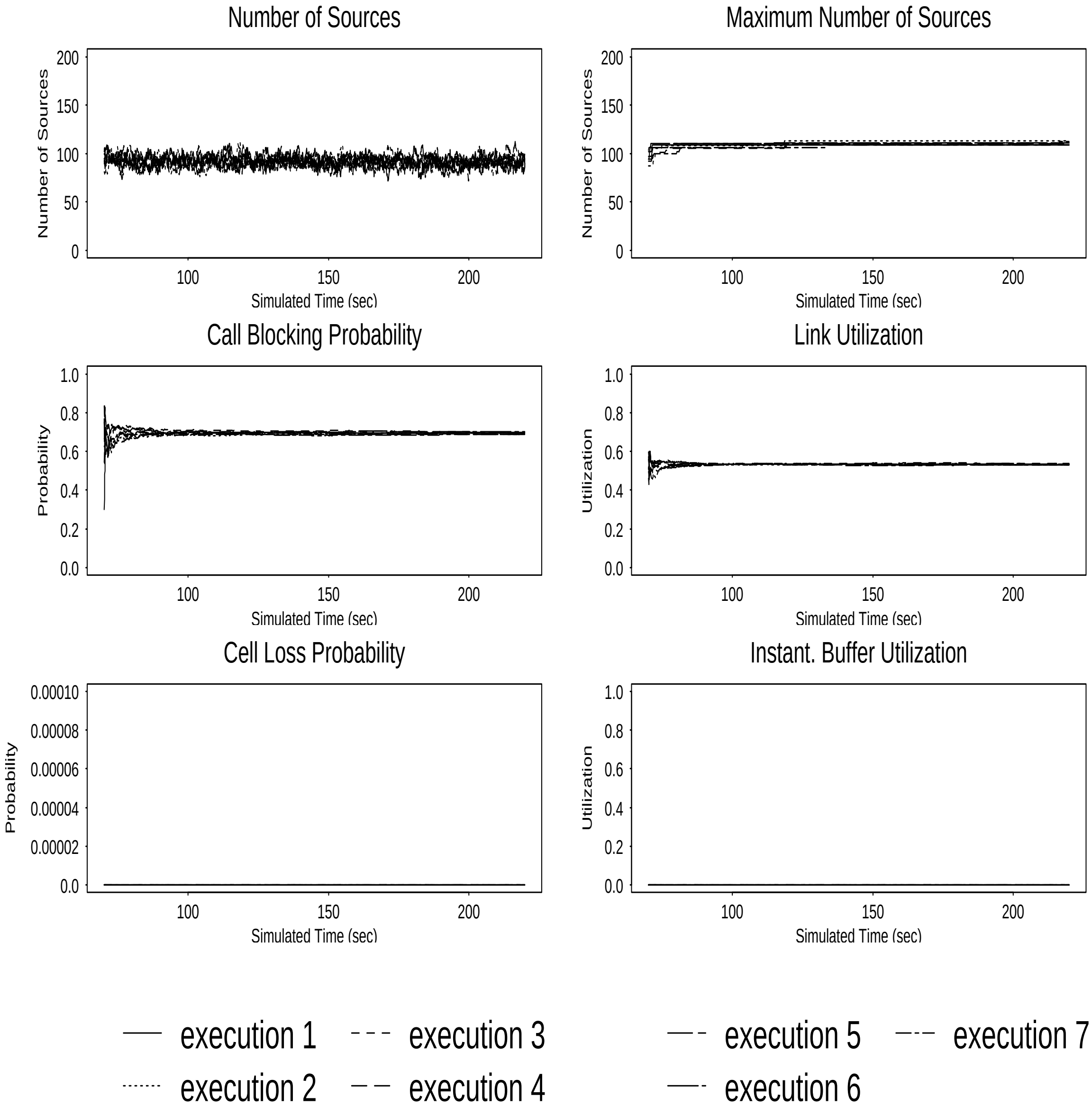

Figure 3.22: Dynamic CAC ($M = 25, r = 20$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds

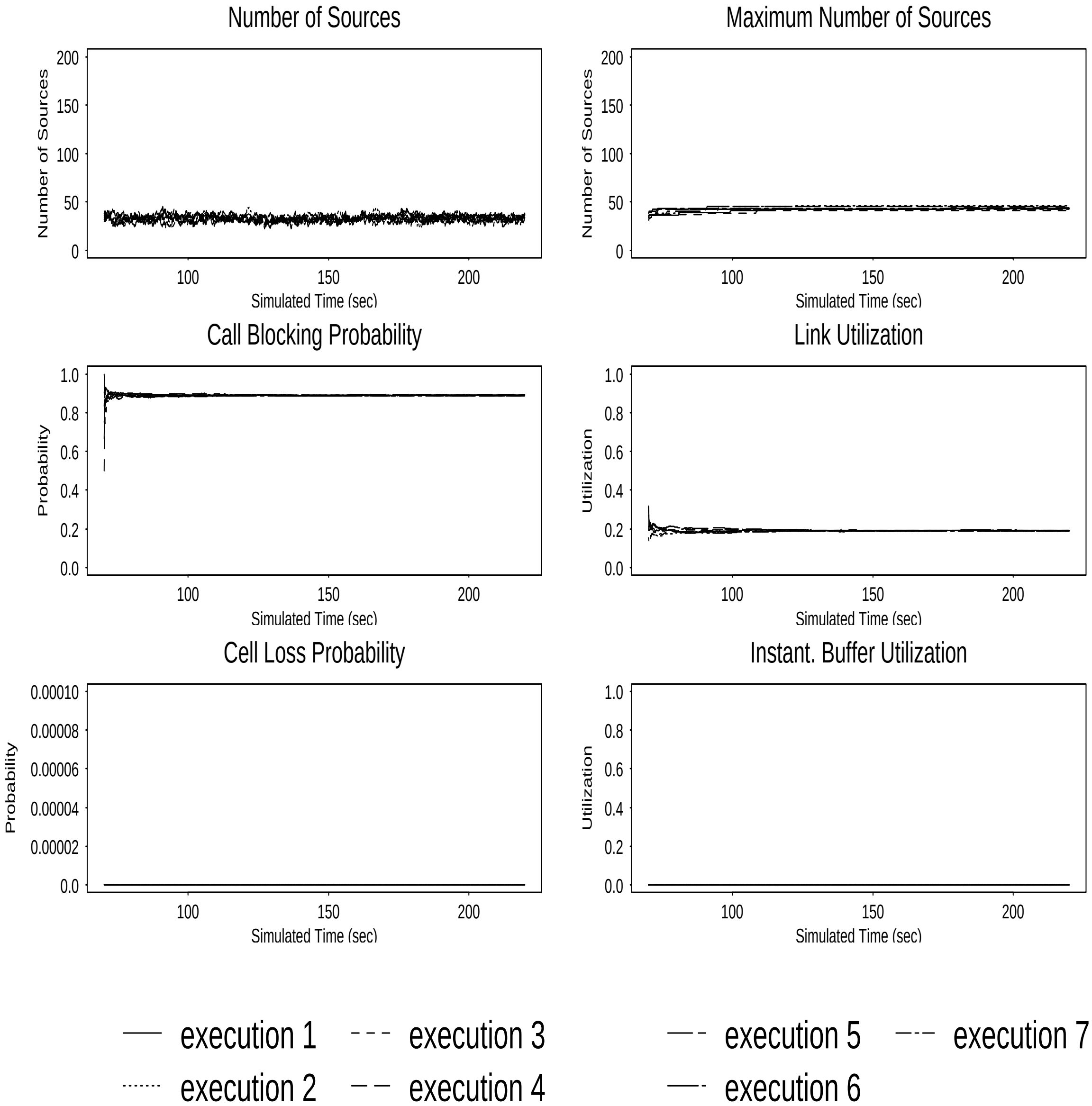

Figure 3.23: Dynamic CAC ($M = 100, r = 5$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds

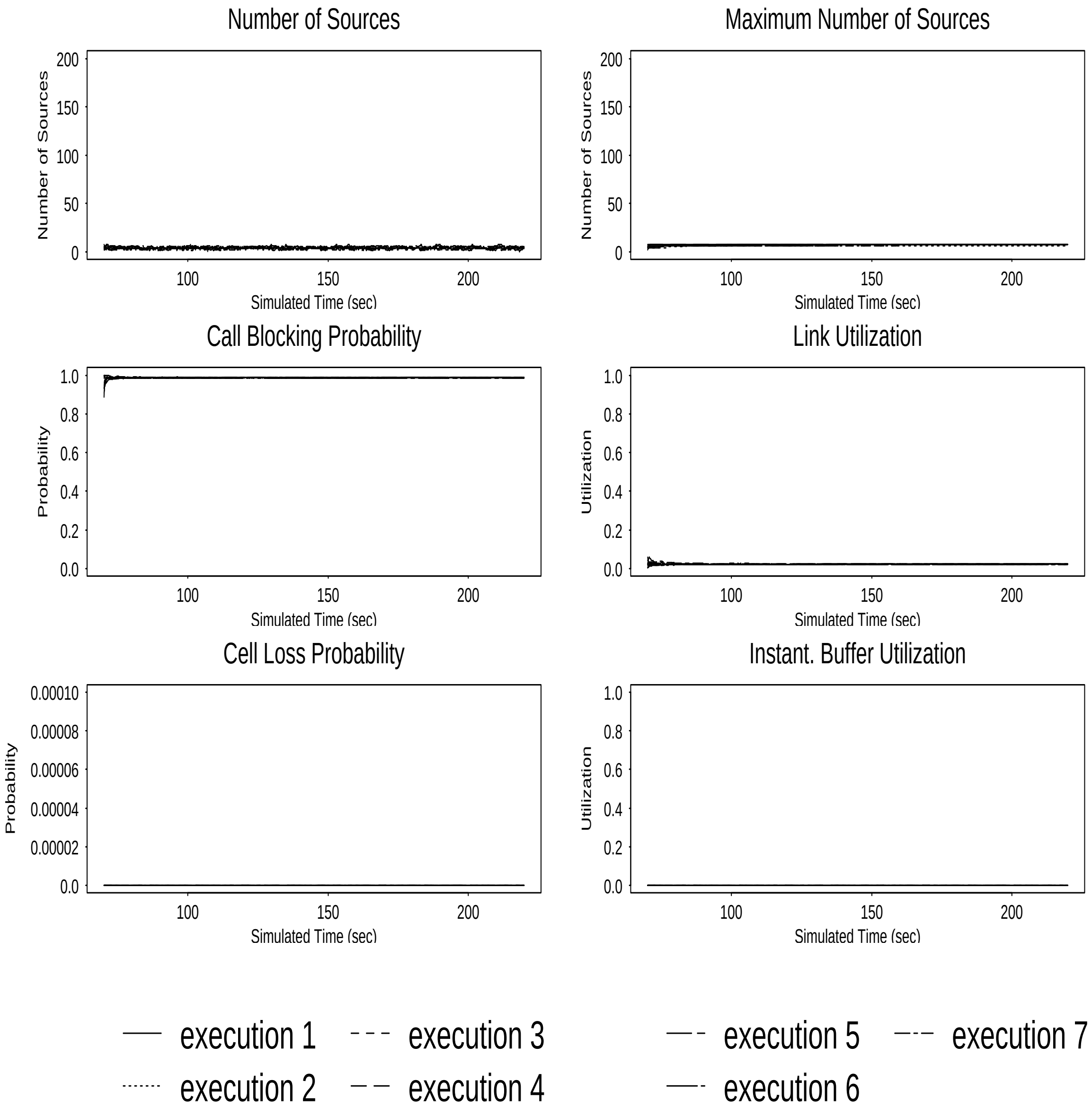

Figure 3.24: Dynamic CAC ($M = 500, r = 1$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds

3.5 Overspecificaton in SCR

By design, the sustained cell rate (SCR) of a source is the upperbound of its mean cell rate. This traffic descriptor is difficult to measure exactly and its declared value may be larger than its actual value. From the view of network administrators fewer calls will be accepted in this case. Moreover, use of SCR as a surrogate for the mean rate will also reduce the efficiency of resource utilization.

The static and Dynamic CAC methods were simulated with the assumption that the peak rate and mean burst length were declared exactly. The peak rate is often a physical hardware limitation so the assumption appears not unreasonable. It is unclear how well mean burst length might be measured if it were ever included as an ATM traffic descriptor.

To investigate overspecification of SCR, the declared utilization was taken to be:

$$\text{declared utilization} = \Delta \times \text{actual utilization}\ ,\ \Delta \geq 1$$

This is equivalent to

$$\text{SCR} = \Delta \times R_{mean}$$

where R_{mean} is the source's actual mean rate, when the peak rate is exact.

Effective bandwdith CAC with and without the Gaussian approximation (Figures 3.25 – 3.26) and Dynamic CAC (Figure 3.27) were simulated with seven executions for each. The value Δ ranged from 1.0 to 2.0. For Dynamic CAC the parameters $\alpha = 0.1, M = 1, r = 500$ were chosen, based on their previous effectiveness (Figure

3.11). Since the simulated version of Dynamic CAC doesn't use SCR it is entirely unaffected by Δ. Link utilization was consistently just over 72%. The other two show decreasing numbers of active sources and link utilization with increasing Δ. If the parameters can be chosen well, Dynamic CAC can perform much better than the other two methods when SCR is overestimated.

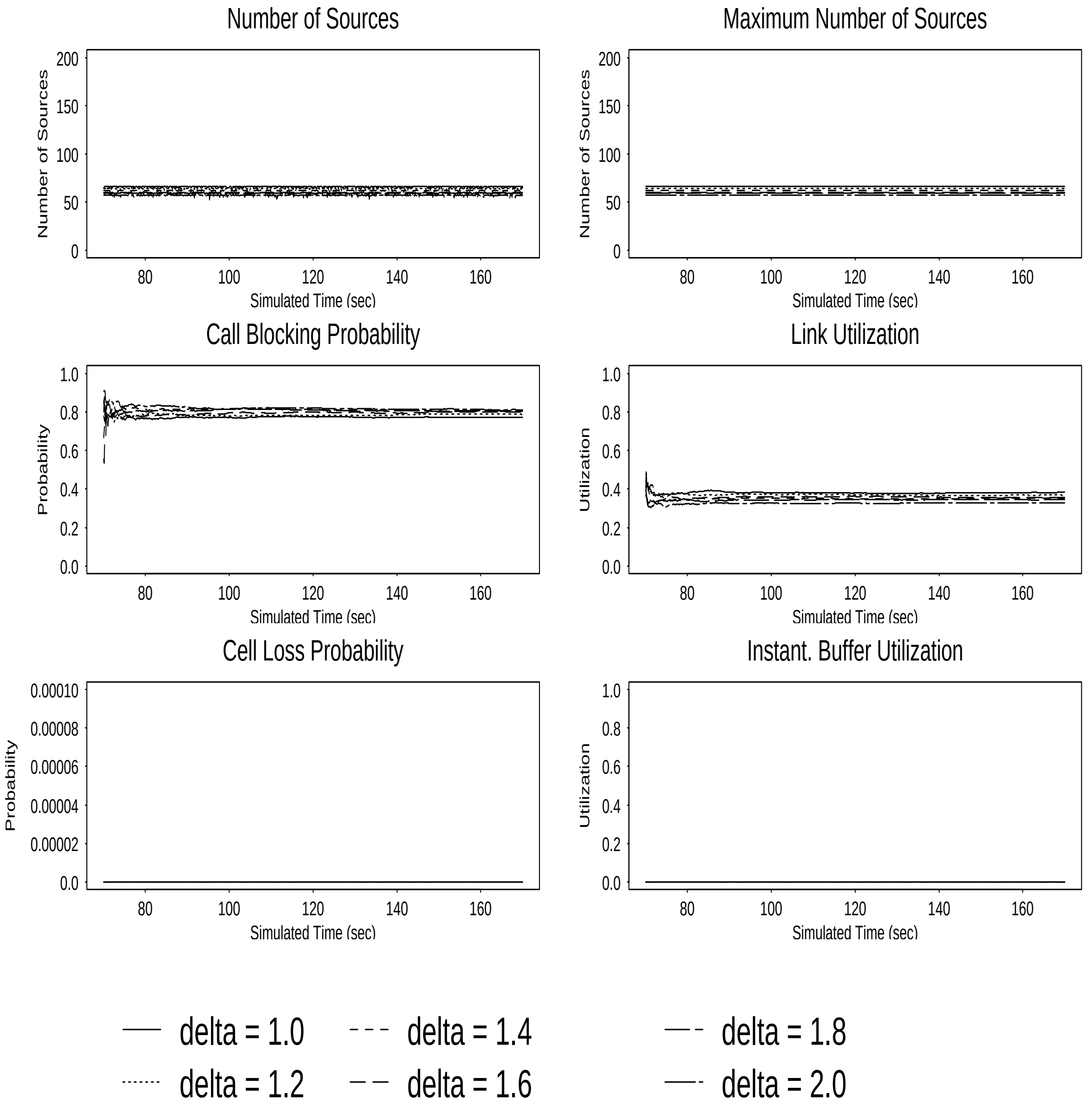

Figure 3.25: Effect of overestimation of SCR on Effective Bandwdith CAC for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds, mean arrival rate 100 calls per second

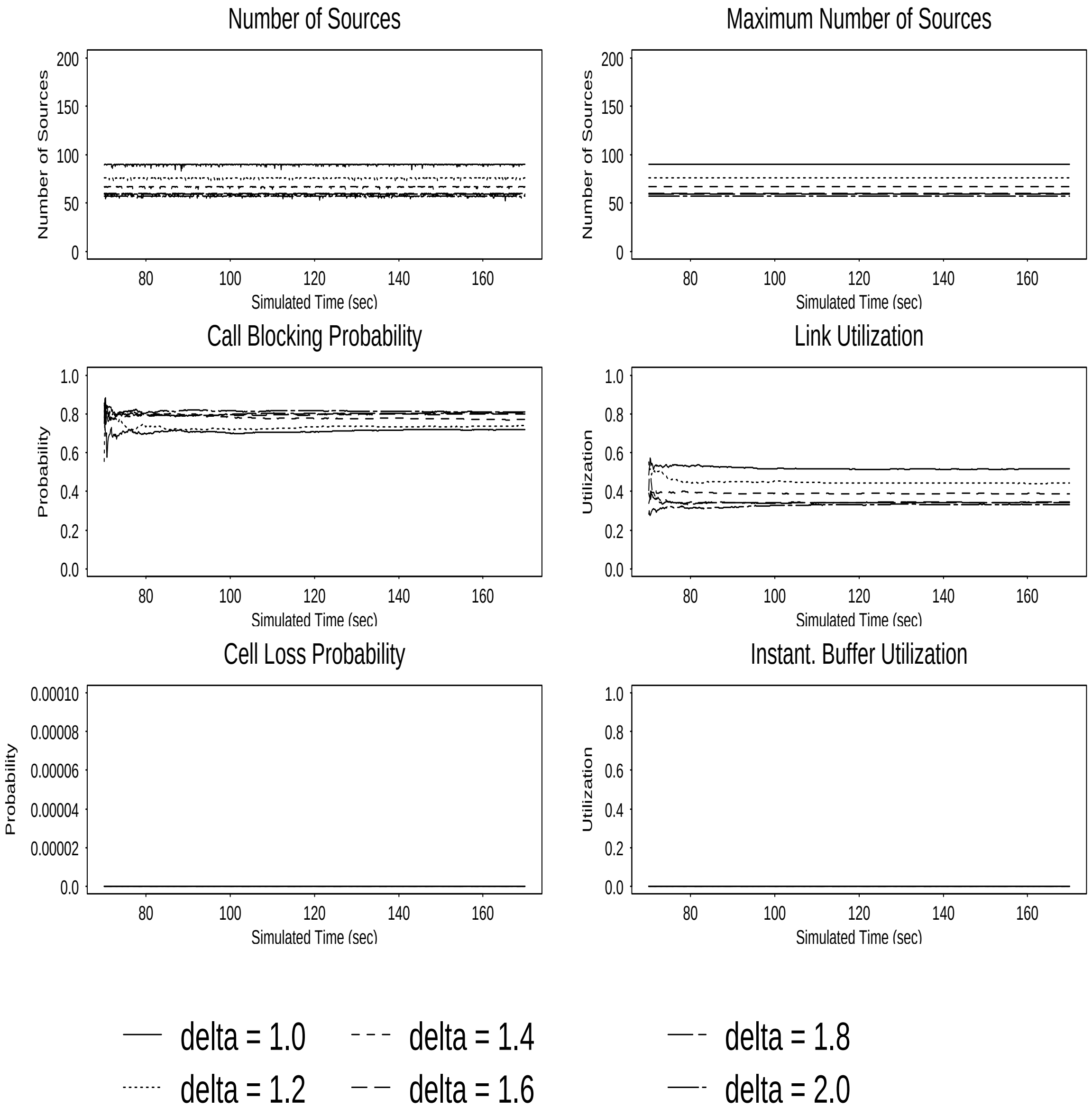

Figure 3.26: Effect of overestimation of SCR on Effective Bandwidth plus Gaussian approximation CAC for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds, mean arrival rate 100 calls per second

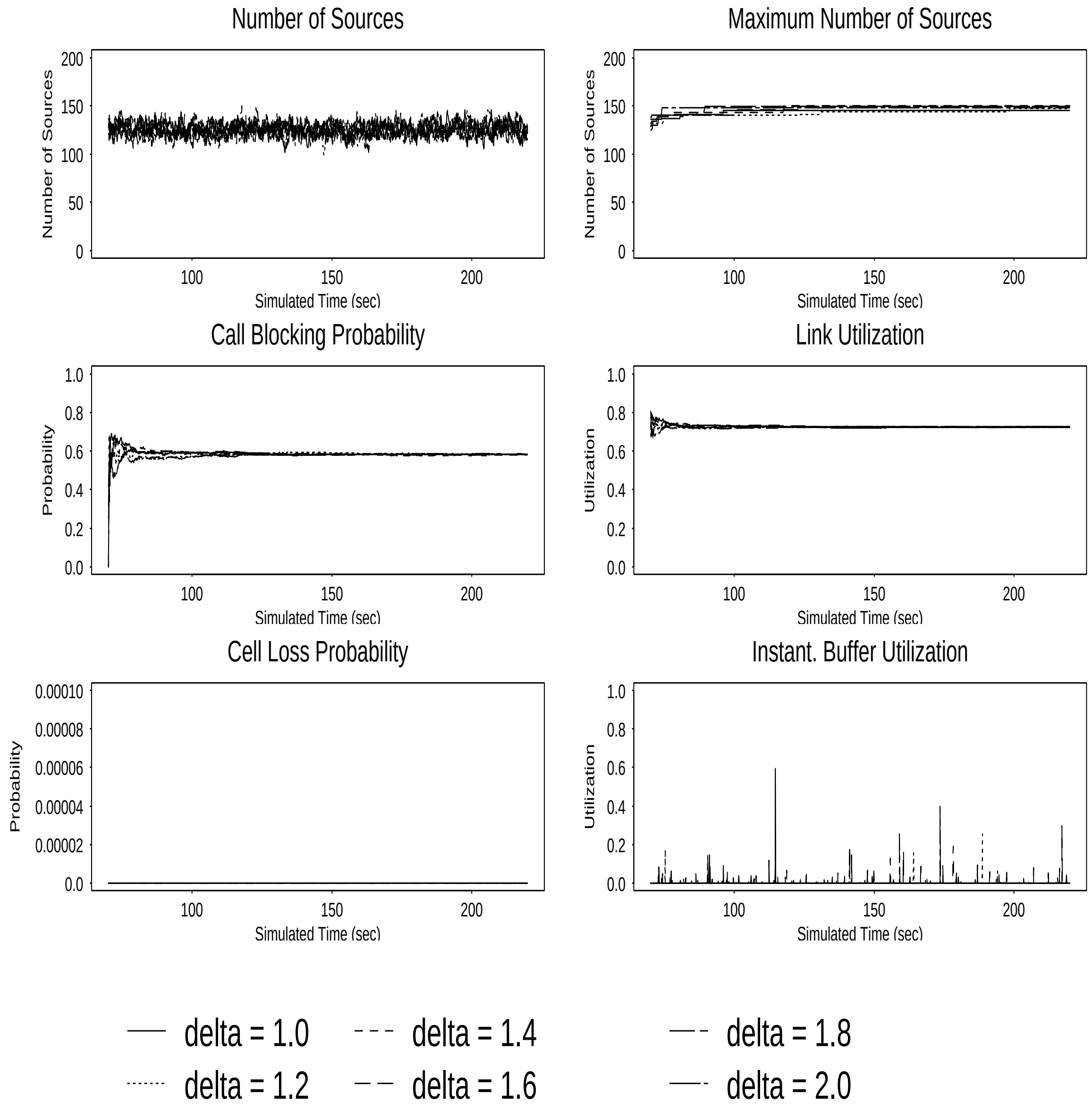

Figure 3.27: Effect of overspecification of SCR on Dynamic CAC ($\alpha = 0.1, M = 1, r = 500$) for On-Off sources, 7075 cell buffer, 10^{-6} overflow probability, mean holding time 3 seconds, mean arrival rate 100 calls per second

Chapter 4

Dynamic CAC with Video Sources

4.1 Star Wars 'Slice' Dataset

To explore the performance of Dynamic CAC with VBR video sources the Star Wars data from Bellcore was obtained [15]. This data provides the number of bytes required per frame ($\frac{1}{24}$ sec) (Figure 4.1) and per slice ($\frac{1}{24\times30}$ frames) to transmit the movie 'Star Wars'. A simplified coding technique, similar to JPEG, was used to permit coding the entire movie. The digitized frame (480 lines by 504 pixels) was partitioned into blocks of 8x8 pixels on which a Discrete Cosine Transform was performed. The coefficients from the transform were uniformly quantized into 8 bits. Huffman coding was then used. No differential or motion compensation coding was used, and only the luminance component (resulting in monochrome video) was used. This data is appealing because it is freely accessible, studied by others, and provides fine detail

(Note: 1 slice $\approx$ 1.4 ms). The data is provided as 60 binary files covering 2850 frames each. For the simulations, slice data from files 15 to 44 was used. This corresponds to frames 39901 to 125400 and inclues the two largest frame sizes. The largest frame requires 78459 bytes which means a peak rate of 15,064,128 bps. This data set was studied in [5] and shown to exhibit long-term dependence with Hurst parameter $H = 0.8 \pm 0.1$.

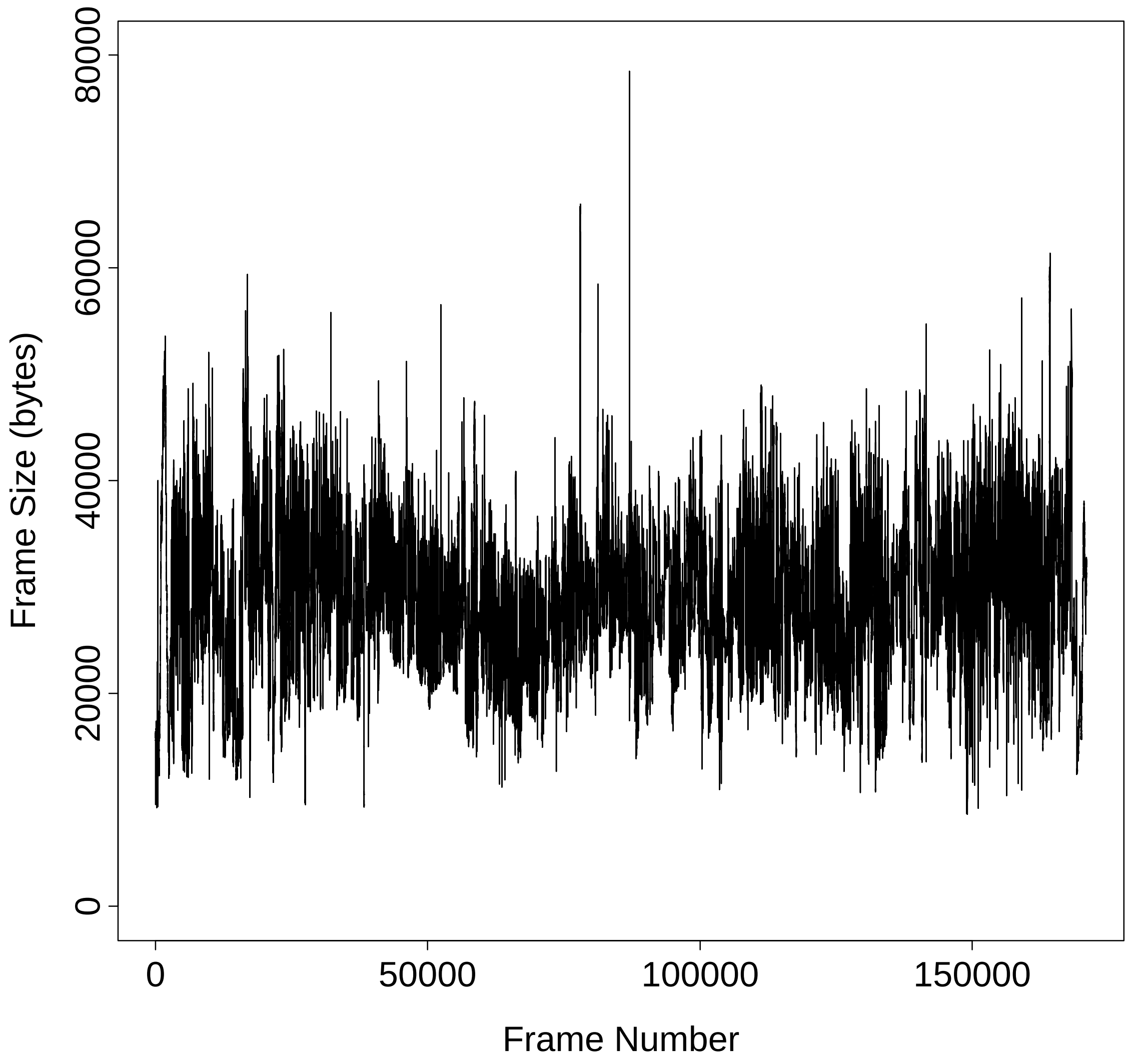

Figure 4.1: Frame Size for Monochrome Star Wars Data

4.2 Overview of the Simulations

To simulate Dynamic CAC with the Star Wars data the Dynamic CAC on-off program was modified. In contrast with on-off sources where the peak rate is fixed and the lengths of the burst and idle periods vary, here the rate per slice varies to accomodate the data for each fixed-length slice. The sending rate for each slice was adjusted, and the time between cells calculated, to produce the appropriate number of evenly spaced cells with the slice. The assumption of regularly-spaced cells is not new ([20], [21]). This number was found based on a payload size of 48 bytes per cell to simulate the overhead introduced by ATM cell headers. Other possible overhead was ignored. Thus, the declared peak rate was taken to be

$$\frac{53}{48} \times 15,064,128 = 16,633,308$$

To simulate additional sources the data for each new source was taken to be some random number of slices between 90 and 150 seconds ahead of the last admitted source. Calls progressing past the end of the data wrapped to the beginning. This technique was used in [20] with 30 minutes of video teleconference data.

All simulations for the 'Star Wars' data were run on the Pentium-Pro. Since no processing was dedicated to generating the traffic data these simulations ran much faster than those for the On-Off sources. One execution, simulating 3500 seconds, had a run-time of approximately 4 hours.

4.2.1 Results of the Simulations

Initial attempts to use Dynamic CAC with the Star Wars slice data showed high cell loss. The first attempt used a simulated run time of 1500 seconds and parameters $M = 1$, $\alpha = 0.1$, and $r = 500$. (Figure 4.2). Calls arrived with mean intensity 100 per second, with a fixed holding time of 1000 seconds. The fixed holding times is evident in the 'staircase' behaviour in the number of sources. Desired overflow was set high at 10^{-6} with a buffer of 7075 cells. Both of these values were reduced in later runs. With these parameters approximately 5.13×10^8 cells were simulated, with an actual cell loss probability of about 2.81×10^{-3}. A second attempt with $r = 1000$ also shows high cell loss (Figure 4.3). Admisson of too many sources during times of low activity may be responsible for the high cell loss. It is interesting to note how the transistion from 24 to 25 sources affected the buffer utilization (and presumably buffer overflow).

To try bridging periods of low activity, $\alpha = .01$ was attempted (Figure 4.4). The difference in cell loss was small, but the difference in the buffer utilization, caused by one less source is striking. In the same vane $\alpha = 0.001$ was successful in that no cell loss was reported (Figure 4.5). Utilization was about 80% on approximately 4.20×10^8 cells. Low buffer utilization suggested a smaller buffer would be sufficient. Thus, the buffer size was reduced to a more reasonable 1000 cells (or about 2.83 ms) (Figure 4.6). Again no cell loss was reported.

The fast execution time of the simulator using video data allowed simulating 3500 seconds for desired overflow 10^{-6}, 10^{-7} and 10^{-8} (Figures 4.7 – 4.9). These correspond to 9.67×10^8, 9.48×10^8 and 9.23×10^8 actual cells respectively. Results for 10^{-8} overflow should be considered preliminary since the actual number of cells is slightly small for this order of magnitude. Low buffer utilization suggests that cell loss is unlikely over a longer run.

The suitability of these parameters for real traffic depends, in part, on the network management scheme. As with On-Off sources, $\alpha = 0.001$ would seem to be a problem if long duration video sources were mixed with shorter calls with similar QoS requirements. Future work would investigate other values of M and r that might allow larger values for α.

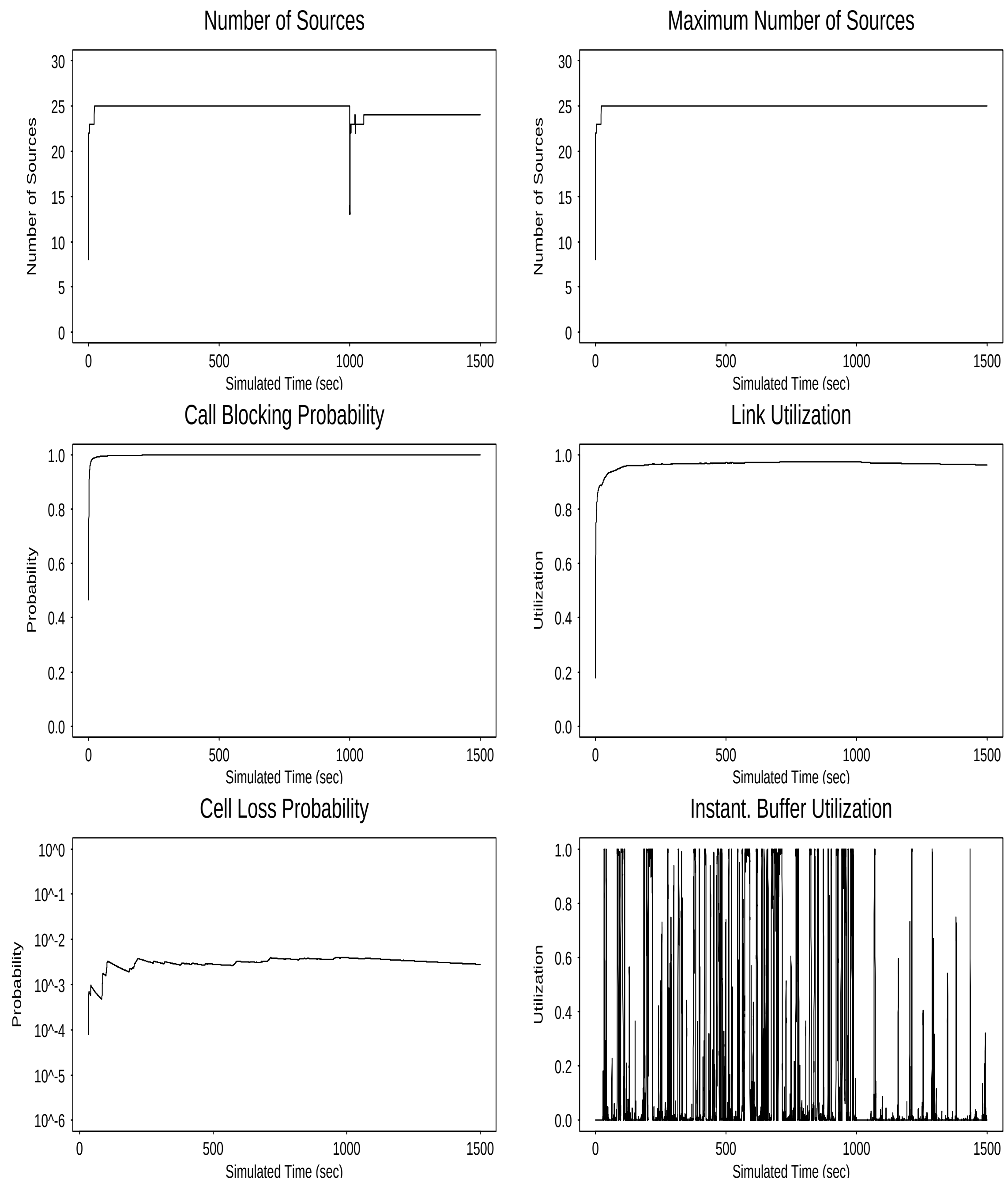

Figure 4.2: Dynamic CAC ($\alpha = 0.1, M = 1, r = 500$), 7075 cell buffer, 10^{-6} overflow probability

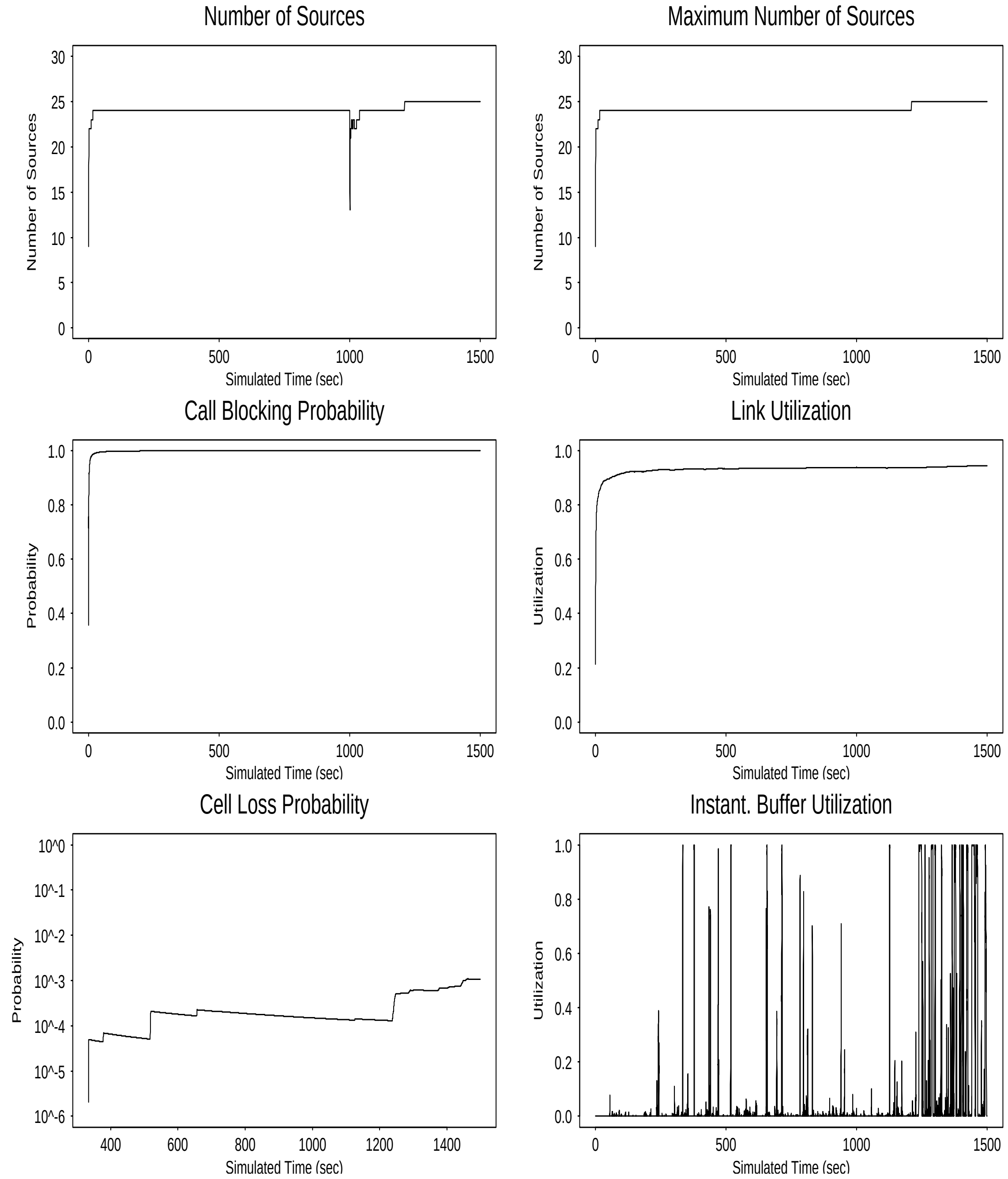

Figure 4.3: Dynamic CAC ($\alpha = 0.1, M = 1, r = 1000$), 7075 cell buffer, 10^{-6} overflow probability

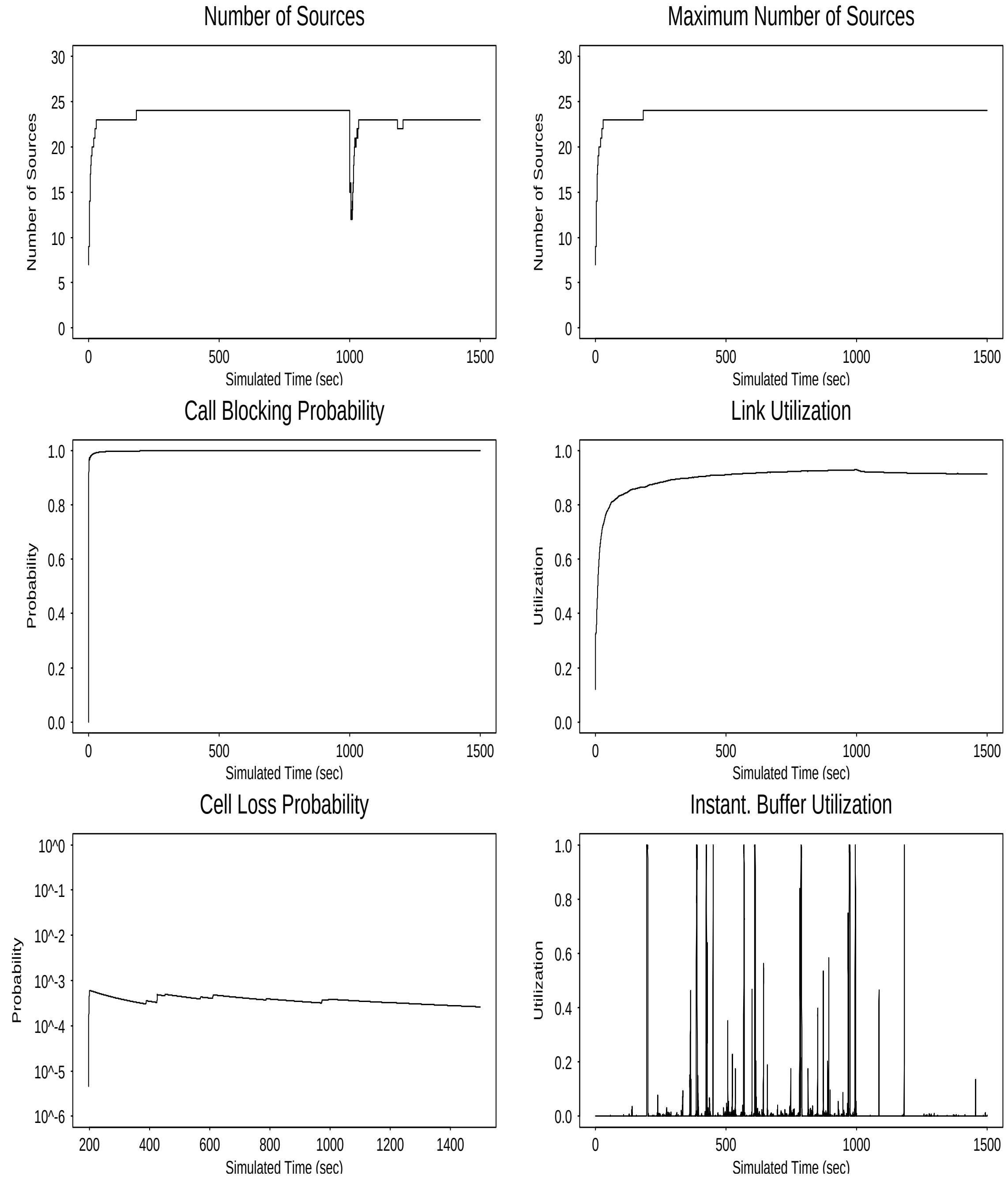

Figure 4.4: Dynamic CAC ($\alpha = 0.01, M = 1, r = 1000$), 7075 cell buffer, 10^{-6} overflow probability

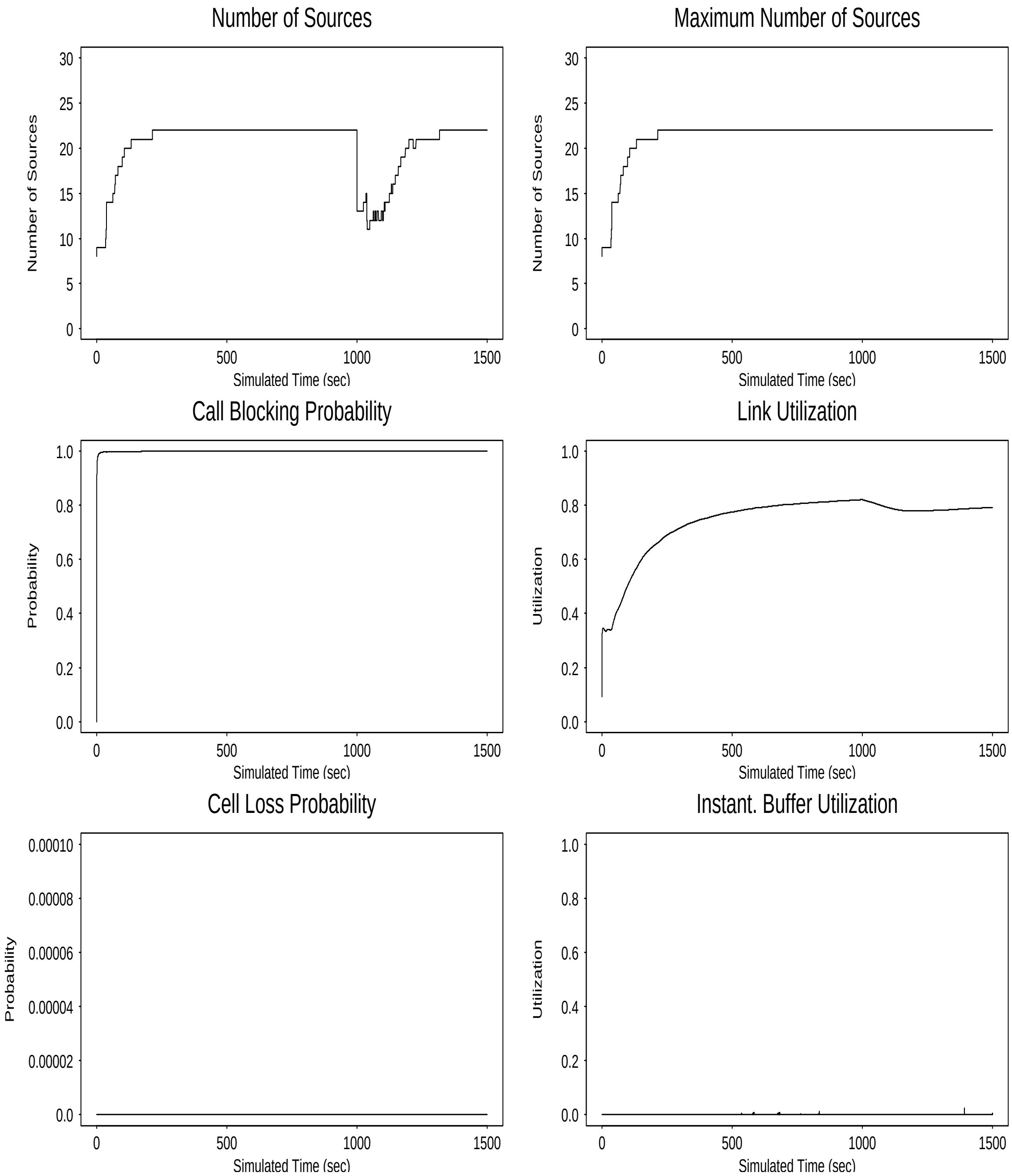

Figure 4.5: Dynamic CAC ($\alpha = 0.001, M = 1, r = 1000$), 7075 cell buffer, 10^{-6} overflow probability

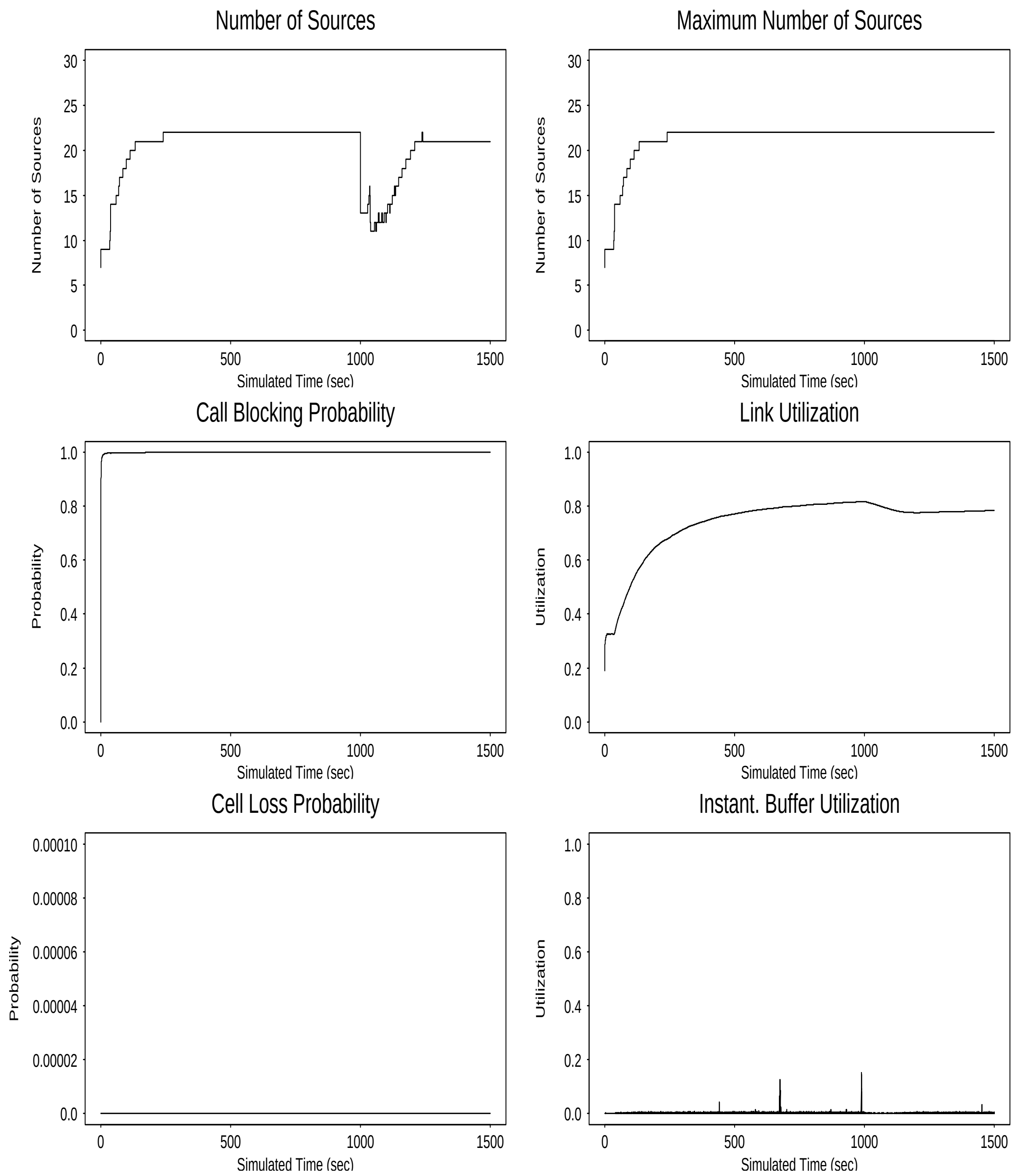

Figure 4.6: Dynamic CAC ($\alpha = 0.001, M = 1, r = 1000$), 1000 cell buffer, 10^{-6} overflow probability

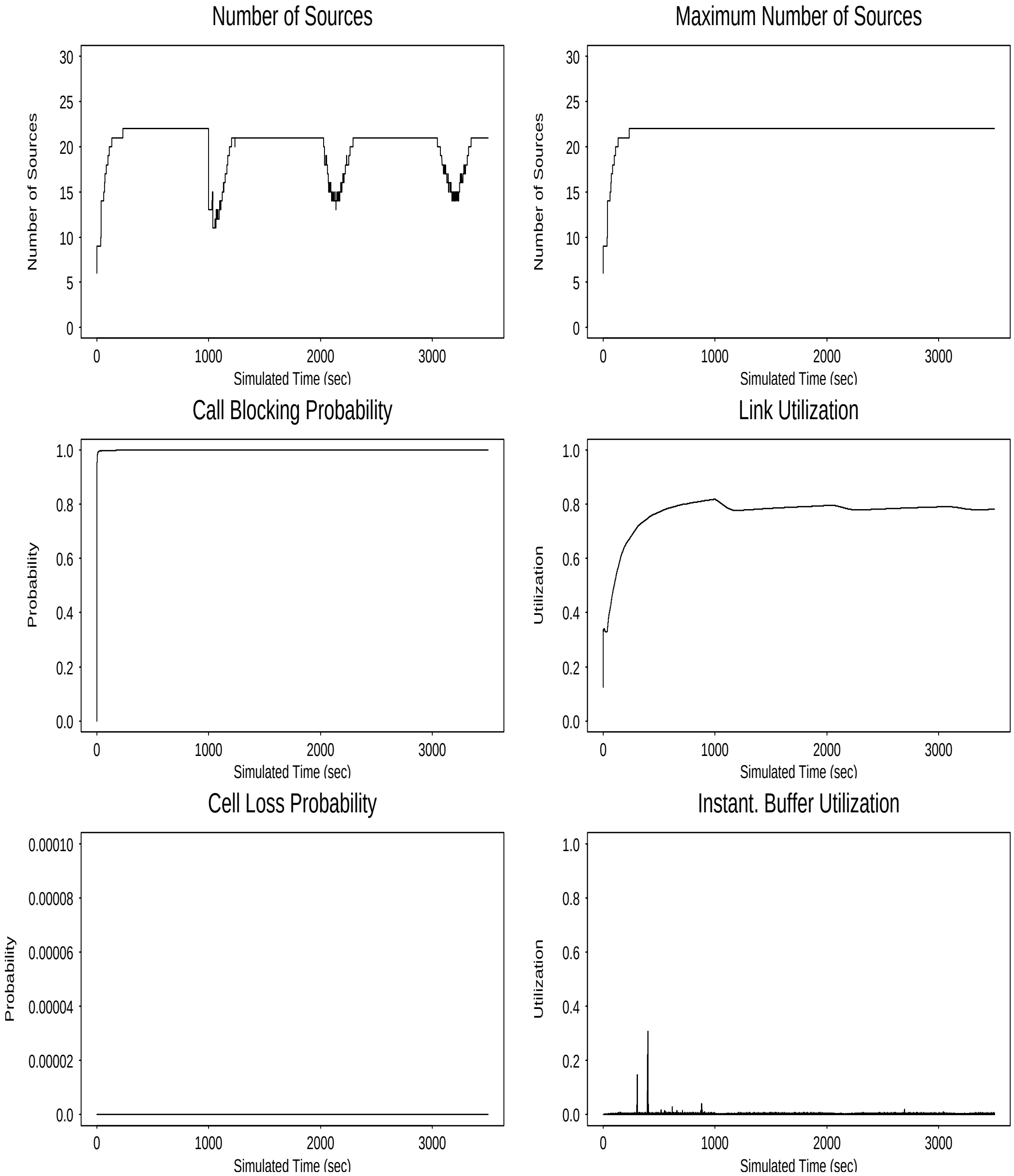

Figure 4.7: Dynamic CAC ($\alpha = 0.001, M = 1, r = 1000$), 1000 cell buffer, 10^{-6} overflow probability

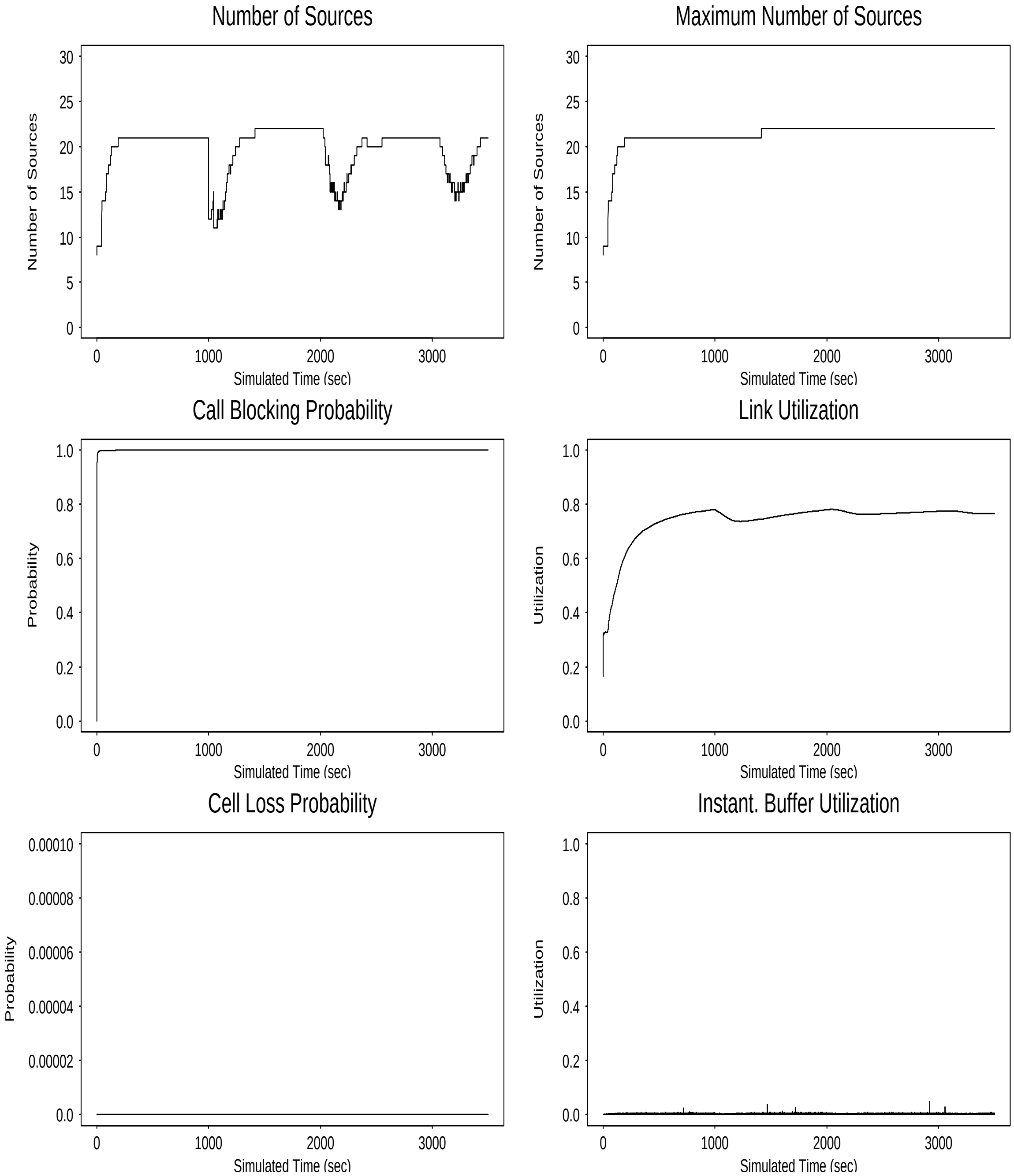

Figure 4.8: Dynamic CAC ($\alpha = 0.001, M = 1, r = 1000$), 1000 cell buffer, 10^{-7} overflow probability

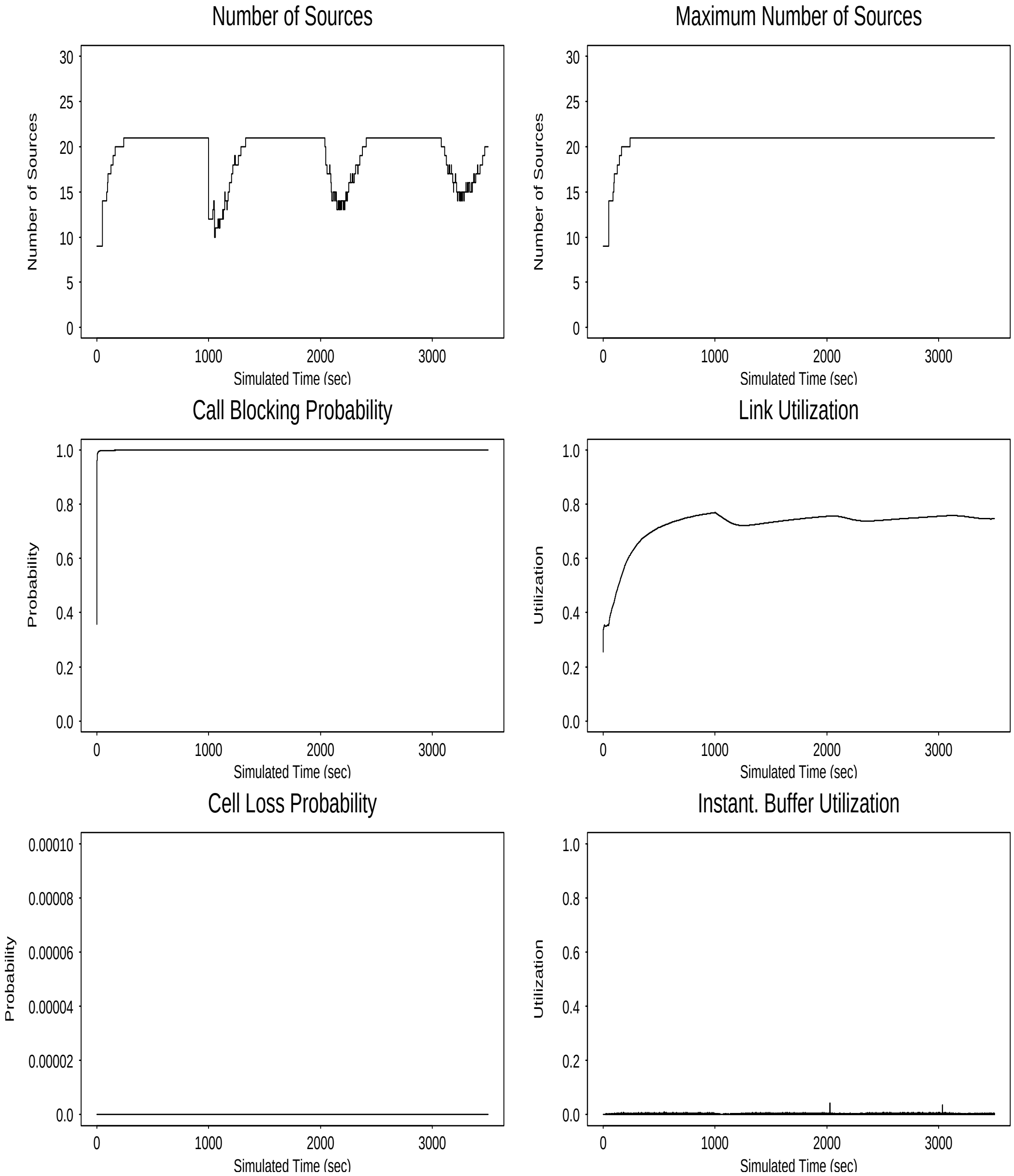

Figure 4.9: Dynamic CAC ($\alpha = 0.001, M = 1, r = 1000$), 1000 cell buffer, 10^{-8} overflow probability

Chapter 5

Conclusions

5.1 Summary of Results

Dynamic CAC is a measurement-based CAC method. Thus, it is vulnerable to over-admission of sources in periods of low activity. Three parameters (M, r, and α) must be set for the operation of Dynamic CAC. In Chapter 3 the effect of these parameters with simulated On-Off sources is investigated. When they are not set properly, the method is not conservative enough. For some values of r too many calls are admitted and cell loss occurs. Larger values of M can prevent cell loss in some of these cases. When r is too small or too big the method is needlessly conservative resulting in low utilization.

Useful values of α are related to the length of the call itself through the exponential decay of old information in the cell arrival distribution. Large values are helpful for bridging periods of low activity, but can result in low utilization when call holding times are short. Large values of α weight the observations of the cell arrivals higher. These admit too many sources and cell loss results. This is likely due to over-admission in periods of low activity.

When its parameters are set well Dynamic CAC can admit more sources than effective bandwidth or effective bandwidth plus Gaussian approximation CAC. Higher link utilizations with no cell loss result. In the case where the sustained cell rate is overspecified the performance of these static CAC methods suffers. Dynamic CAC is unaffected by this overspecification.

For video sources, Dynamic CAC works well, even when a reference model is unknown. Again, correct choice of parameters is important. Longer holding times associated with video sources allow smaller values of α. This allows bridging periods of low activity.

Sensitivity to Dynamic CAC parameters suggest some problems will be encountered with heterogeneous traffic. For example, short holding times are problematic with the small values of α which are useful for bursty sources like video.

5.2 Areas for Future Work

The results of this project suggest several areas for future study. Depending on the network management scheme, it may be desirable to combine long video calls and short calls of unknown type. Useful values of M, r and α are needed for such a mix of traffic.

The relationship between values of α and call holding times suggest that a different method for combining the cell arrival distribution and the current observations of the cell arrival process would be helpful.

A new bound on the cell loss probability may improve the performance of Dynamic CAC by reducing sensitivity to parameters. Such a bound might even include the buffer size explicitly.

A Dynamic CAC algorithm with some policy for partitioning the buffer should be explored. This would be useful for hetergeneous traffic sources or widely varying CLR requests. This would help deal with the problems associated with sensitivity to parameters that may arise with diverse traffic types.

Real ATM networks will exhibit cell delay variation. Its affect on the usefulness of counting the numbers of arriving cells in a measurement period should be investigated.

Bibliography

[1] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic Theory of a Data–Handling System," *Bell Syst. Tech. Journal*, vol. 61, pp. 1871-1894, Oct. 1982.

[2] ATM Forum, *ATM User-Network Interface Specification: Version 3.1,* 1994.

[3] ATM Forum, "ATM Service Categories: The Benefits to the User," *Library White Papers*, April 1997.

[4] J. Bae and T. Suda, "Survey of Traffic Control Schemes and Protocols in ATM Networks," *Proc. of the IEEE*, vol 79, pp. 170-189, Feb. 1991.

[5] J. Beran, R. Sherman, M. Taqqu and W. Willinger, "Long–Range Dependence in Variable-Bit-Rate Video Traffic," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1566-1579,

[6] H. Bruneel and B. G. Kim, *Discrete-Time Models for Communication Systems Including ATM.* Kluwer Academic: Boston 1993.

[7] G. L. Choudhury, D. M. Lucatoni and W. Whitt, "Squeezing the Most Out of ATM," *IEEE Trans. Commun.*, vol. 44, pp. 203-217, Feb. 1996.

[8] J. N. Daigle and J. D. Langford, "Models for Analysis of Packet Voice Communications Systems," *IEEE J. Select. Areas. Commun.*, vol. SAC-4, pp. 847-855, Sept. 1986.

[9] M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN.* 3rd ed., Englewood Cliffs, N.J.: Prentice-Hall 1995.

[10] C. Douligeris and G. Develekos, "Neuro-Fuzzy Control in ATM Networks", *IEEE Communications Magazine,* vol. 35, pp. 154-162, May 1997.

[11] A. I. Elwalid and D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks," *IEEE Trans. Communication*, vol.1, pp. 329-343, June 1993.

[12] A. Elwalid, D. Heyman, T. V. Lakshman, and D. Mitra, "Fundamental Bounds and Approximations for ATM Multiplexers with Applications to Video Teleconferencing," *IEEE J. Select. Areas. Commun.*, vol. 13, pp. 1004-1016, Aug. 1995.

[13] R. Engel and J. S. Turner, "Real-Time Connection Admission Control with Multistate Traffic Sources," *Proc. ICC 1997.*

[14] A. Erramilli, O. Narayan and W. Willinger, "Experimental Queueing Analysis with Long-Range Dependent Traffic," *IEEE/ACM Trans. Networking*, vol. 4, pp. 209-223, April 1996.

[15] M. W. Garrett and W. Willinger, "Analysis, Modelling and Generation of Self-Similar VBR Video Traffic," *Proc. ACM SIGCOMM 1994*, Aug. 1994.

[16] R. J. Gibbens and P. J. Hunt "Effective bandwidths for the multi-type UAS channel," *Queueing Systems*, vol. 9, pp. 17-28, 1991.

[17] R. Guérin, H. Ahmadi and M. Nagshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks," *IEEE J. Select. Areas. Commun.*, vol. 9, pp. 968-981, Sept. 1991.

[18] J. J. Harms and H. Qian, "A Call Admission Scheme for ATM Networks Based on Refinement of Traffic Measures," *Proc. ICC 1995*, pp. 191-195.

[19] H. Heffes and D. M. Lucatoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *IEEE J. Select. Areas. Commun.*, vol. SAC-4, pp. 856-869, Sept. 1986.

[20] D. P. Heyman, A. Tabatabai and T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks," *IEEE Trans. Circuits. Syst. Video. Technol.*, pp. 49-59, Mar. 1992.

[21] D. P. Heyman and T. V. Lakshman, “Source Models for VBR Broadcast–Video Traffic,” *IEEE/ACM Trans. Networking*, vol. 4, pp. 40-48, Feb. 1996.

[22] D. P. Heyman and T. V. Lakshman, “What Are the Implications of Long- Range Dependence for VBR-Video Traffic Engineering?,” *IEEE/ACM Trans. Networking*, vol. 4, pp. 301-317, June 1996.

[23] P. Joos and W. Verbiest ”A Statistical Bandwidth Allocation and Usage Monitoring Algorithm for ATM Networks,” *Proc. ICC 1989*, pp. 415-422.

[24] S. H. Kang and D. K Sung, “Real-Time Cell Loss Estimation for ATM Multiplexers with Heterogeneous ON/OFF Sources,” *Proc. ICC 1997*, pp.

[25] K. Kawashima and H. Saito, “Teletraffic Issues in ATM Networks,” *Computer Networks and ISDN Systems*, vol. 20, pp. 369-375.

[26] F. P. Kelly, “Effective Bandwidths at Multi-Class Queues,” *Queueing Systems*, vol. 9, pp. 5-16, 1991.

[27] G. Kesidis, J. Walrand, and C. Chang, “Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources,” *IEEE Trans. Communication*, vol. 1, pp. 424- 428, Aug. 1993.

[28] L. Kosten, “Liquid Models for a Type of Information Buffer Problems,” *Delft Progess Report*, vol. 1, pp. 71-86, 1986.

[29] K. R. Krishnan and G. Meempat, "Long–range dependence in VBR video streams and ATM traffic engineering," *Performance Evaluation*, vol 30, pp. 45-56, 1997.

[30] K.R. Krishnan, "A new class of performance results for fractional brownian traffic model", *Queueing Systems*, vol. 22, pp. 277-285, 1996.

[31] T. Lee, K. Lai and S. Duann, "Design of a Real-Time Call Admission Controller for ATM Networks," *IEEE/ACM Trans. Networking*, pp. 758-765, Oct. 1996.

[32] R. S. Lehman, *Computer Simulation and Modeling: An Introduction*. Laurence Erlbaum Associates: Hillsdale N.J. 1977.

[33] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version), *IEEE/ACM Trans. Networking*, vol. 2, pp. 1-15, Feb. 1994.

[34] K. Liu, D. W. Petr and C. Braun, "A Measurement-Based CAC Strategy for ATM Networks," *Proc. ICC 1997.*

[35] K. Liu *et.al.*, "Design and Analysis of a Bandwidth Management Framework for ATM-Based Broadband ISDN", *IEEE Communications Magazine*, vol. 35, pp. 138-145, May 1997.

[36] B. Maglaris, C. Anastassiou, P. Sen, G. Karlsson and J. D. Robbins, "Performance Models of Statistical Multiplexing in Packet Video Communications," *IEEE Trans. Communication*, vol. 36, pp. 834-843, July 1988.

[37] M. Masugi, "Cell Loss Rate Estimation based on Neural Network for Call Admission Control in ATM Networks," textslProc. ICC 1997.

[38] A. Mohammadi, S. Kumar and D. Klymyshyn, "Characterization of Effecitive Bandwidth as a Metric of Quality of Service for Wired and Wireless ATM Networks," *Proc. ICC 1997.*

[39] J. A. S. Monteiro, M. Gerla and L. Fratta, "Statistical Multiplexing in ATM Networks," *Proc. 4th Int. Conf. On. Data Communication Systems and Their Performance,* Barcelona, Spain, pp. 187-201.

[40] C. Liu and H.T. Mouftah, "Virtual Call Admission Control– A Stategy for Dyanamic Routing over ATM Networks," *Proc. ICC 1995.*

[41] T. Murase, H. Suzuki and T. Takeuchi, "A Call Admission Control Scheme for ATM Networks Using a Simple Quality Estimate," *IEEE J. Select. Areas. Commun.*, vol. 9, pp. 1461-1470, Dec. 1991.

[42] I. Norros, "On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks," *IEEE J. Select. Areas. Commun.*, vol. 13, pp. 953-962, Aug. 1995.

[43] H. Park, D. Kwak, W. Rhee, M. Jeon and J. Kim, "Adaptive CAC Alogirthms Using Measurement and Estimation," *Proc. ICC 1995*, pp. 211-215.

[44] K.W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*. New York: Springer 1995.

[45] H. Saito and K. Shiomoto, "Dynamic Call Admission Control in ATM Networks," *IEEE J. Select. Areas. Commun.*, vol. 9, pp. 982-989, Sept. 1991.

[46] H. Saito, "Hybrid Connection Admission Control in ATM Networks," *Proc. ICC 1992*, pp. 699-703.

[47] H. Saito, "Call Admision Control in an ATM Network Using Upper Bound of Cell Loss Probability," *IEEE J. Select. Areas. Commun.*, vol. 40, pp. 1512-1521, Sept. 1992.

[48] H. Saito, "Dynamic Resource Allocation in ATM Networks," *IEEE Communication Magazine*, vol. 35, pp. 146-153, May 1997.

[49] B. Tsybakov and N. D. Georganas, "Overflow Probability In an ATM Queue with Self-Similar Input Traffic," *Proc. ICC 1997*.

[50] C. Yang and A. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks", *IEEE Network*, vol. 9, pp. 34-35, July/Aug. 1995.

Vita

Name	David A. Rolls
Education	M.Sc. (In Progress) Queen's University B.Ed. (June 1993) University of Western Ontario B.Math(Honours) (May 1993) University of Waterloo
Experience	Instructor(Math 128J), Queen's University, Kingston, Ontario, Winter 1997 Instructor(English), ECC Foreign Language Institute, Osaka, Japan, 1994 – 1996 Teacher (High School Mathematics), Halton Region Board of Education, Burlington, Ontario, Sept. – Dec. 1992 Teacher (High School Mathematics), North York Board of Education, North York, Ontario, Jan. – April 1992 Teacher Intern (High School Mathematics), Carleton Board of Education, Nepean, Ontario, Jan. – April 1991
Awards	Queen's Graduate Award, Sept. 1996 Suncor Award for Scientific Excellence, May 1987
Affiliations	Ontario College of Teachers, AMS, IEEE