Applying Stochastic Network Calculus In Scenarios With Incomplete Knowledge

Autor:
Sebastian Henningsen

Erstprüfer:
Prof. Dr.-Ing. Jens B. Schmitt

Zweitprüfer:
Dipl.-Math. Michael Beck

29.1.2013
Abstract

The deterministic network calculus (DNC) and its probabilistic counterpart stochastic network calculus (SNC) are promising theories which provide methodologies to analyse and design networked systems. As the “little brother” of queuing theory, the main goal of network calculus has always been the well-founded design of future networks, since its origins in the 1990’s [17]. Therefore, the analysis of a network with (S)NC is always static as it is aimed to provide (performance) guarantees for all points in time. Moreover, a complete knowledge about the system topology and structure of arrival flows is necessary in order to provide the desired bounds. However, in real-world scenarios the structure of arrivals is, in contrast to the topology, hardly known.

In order to fill this gap, in this thesis a first approach for arrival estimation is proposed and interleaved with the traditional performance bounds. Furthermore, a more dynamic view on systems is introduced and compared to the classical point of view. As this work is a first advance on dynamic system analysis with (S)NC, exponentially distributed arrivals are assumed throughout this work, due to reasons of simplicity.

The rest of this work is structured as follows. First, a motivation and a short introduction to stochastic network calculus is given. In the second chapter the arrival flow estimator is proposed and woven into the known performance bounds. The third chapter enriches the theory with simulated evaluation and comparison to the classic model. Moreover, some assumptions will be dropped which leads to the loss of formal guarantees (i.e. the performance bounds do not hold), however the simulation results will save the day. In chapter four some related work is covered, additionally some directions for future work will be discussed, such as the generalization of the theoretical results from chapter two. In the end, everything is wrapped up in the conclusion.
## Contents

1 Introduction 3  
1.1 Motivation 3  
1.2 Introduction to Stochastic Network Calculus 4  

2 Theoretical Framework 9  
2.1 Estimating the distribution parameter 9  
2.2 Performance Bounds 12  

3 Evaluation 19  
3.1 Introduction to Markov Chains 19  
3.2 A Markov Modulated Arrival Process 21  
3.3 Theory vs. Practice 22  
3.3.1 Evaluation of the Parameter Estimator 23  
3.3.2 Comparison to traditional Performance Bounds 27  
3.3.3 A backlog bound over finite Time Horizons 45  

4 Related and Future Work 48  
4.1 Related Work 48  
4.2 Future Work 49  

5 Conclusion 51
1 Introduction

1.1 Motivation

In the field of distributed systems, theoretical frameworks are used as helpful tools for systems design. The probably most prominent example is queuing theory [21], [13] which was build upon the need to design and deploy telephone networks. Despite its enormous success it became obvious, that queuing theory is not the best fit when it comes to the analysis of computer networks, see the introduction of [8].

Furthermore, queuing theory focuses on the average case analysis whereas system engineers (especially of safety-critical systems) are more interested in worst-case performance guarantees. As the “little brother” of queuing theory, network calculus aims at filling this gap. Although most of the bounds derived with DNC are known to be tight (i.e. there is a system state in which these bounds are met with equality), the obtained results are not very realistic. This stems from the problem that even very unlikely events have to be considered in the analysis.

Therefore, the idea of stochastic network calculus (SNC) is to provide stochastic performance guarantees: bounds that hold with a high probability, to be more precise. There are several approaches to a stochastic network calculus, the most prominent ones were developed by Chang [7], Liebeherr et al. [4] and Jiang [16]. In this thesis we will utilize (and extend) the SNC with moment generating functions (MGFs) proposed by Chang.

Traditionally, in (S)NC one has to find a suitable abstraction for all possible arriving flows and the possible service a node offers - this is the concept of arrival and service curves. Whereas finding a suitable service curve is relatively easy, binding the possible arrivals can be quite hard. Even if an arrival curve has been found, a poorly chosen restriction will lead to loose bounds. Therefore, an arrival curve estimator is introduced in this thesis, which outputs an estimated arrival curve based on measured samples of the arriving flow. Furthermore, the uncertainty of the estimator is interleaved with the violation probability of the traditional stochastic performance bounds.

Moreover, consider a system with a single node and some service curve as well as an arriving flow, also bounded by an arrival curve and overall discrete time (i.e. time is a sequence of naturals). Then, when analyzing the system with SNC, one has a very static point of view: Everything (topology, arrival
and service curves) has to be known in advance and it does not play a role if the system is currently running - even if it was, this knowledge could not be utilized in the analysis. Besides, when calculating bounds it is assumed, that the system is running since point in time 0 and everything is computed relatively to this point.

The arrival curve estimator proposed in section 2 can be used to obtain a more dynamic view and exploit the additional knowledge which can be drawn from a running system. Assume the arrival flow samples are measured at every time step and an arrival curve is estimated over a sliding window of fixed size of these samples, then the estimated curve will adapt itself to the current situation. In order to utilize this continuous adjustment one has to “redefine” the datum of time from 0 to the point of the last arrival curve estimation.

Thus, a variety of applications is plausible, the most simple one being a program monitoring a networked system and generating outputs in the form of “In the next $n$ seconds, the backlog will not exceed $b$ bytes with probability $p$”.

1.2 Introduction to Stochastic Network Calculus

This thesis requires some basic knowledge in probability theory and deterministic calculus, for a rigorous introduction to network calculus please refer to the tutorial of Le Boudec and Thiran [17]. In the following a brief introduction to the stochastic network calculus with moment generating function (also called MGF-calculus or $(\sigma(\theta), \rho(\theta))$-calculus) developed by Chang in [7] will be given.

In the deterministic network calculus arrivals at a serving element are characterized by abstract flows. The service element itself is abstracted by the concept of nodes, where a node has an amount of input flows which are processed under certain scheduling policies and in the end relayed as departing flows. These arriving flows are traditionally characterized by cumulative functions, mostly denoted by $A(x)$ which specifies the amount of arrived data/packets/what-is-measured up to point in time $x$. Note that, depending on whether we use a discrete or fluid time model it holds $x \in \mathbb{N}$ or $x \in \mathbb{R}$, respectively. In this thesis we will assume time to be discrete.

In figure 1 the input and output flow at a single node is illustrated, as well as the delay ($d(x)$) and backlog ($x(t)$) at time $t$. 

In the deterministic network calculus (as the name already suggests), flows are not subject to random decisions. In order to generalise this idea, stochastic flows are defined by using random variables but are at the same time very similar to their deterministic counterparts.

**Definition 1.1.** Let \((a(k))_{k \in \mathbb{N}}\) be a sequence of non-negative, real-valued random variables. The stochastic process \(A\) with time space \(\mathbb{N}_0\) and state space \(\mathbb{R}_0^+\) defined as

\[
A(n) := \sum_{i=0}^{n} a(i)
\]

is called a flow and the \(a(i)\) increments of the flow.

The increment \(a_i\) can be interpreted as the amount of data (or whatever is measured) that arrived between the times \(i - 1\) and \(i\). In deterministic network calculus input flows (in particular: every time interval of the input flow) are bounded by so called arrival curves. However, when working with stochastic flows, the deterministic arrival curves do not exploit the benefits of the arrivals’ randomness or worse, do not exist for certain families of stochastic arrival flows as we will see in the following two examples.

Please recall that a flow \(A\) has a deterministic arrival curve \(\alpha\) if it holds \(A(n) - A(m) \leq \alpha(n - m)\) for all \(n \geq m \geq 0\).

For the first example let the increments of the flow be independent Bernoulli distributed, thus it holds for every \(a(i)\) and for all \(p \in (0, 1)\):

\[
\mathbb{P}(a(i) = 0) = p \\
\mathbb{P}(a(i) = 1) = 1 - p
\]
Intuitively, a deterministic bound on $A$ has to assume, that it holds $a(i) = 1$ for each time step, since there is always a small probability, that every random variable up to point $n$ is 1. The following argumentation underlines (and proves) this intuition. We are interested whether the probability for $A(n) - A(m) = n - m$ is larger than 0:

$$P(A(n) - A(m) = n - m) = P(\sum_{i=m+1}^{n} a(i) = n - m)$$

$$= P(\bigcap_{i=m+1}^{n} \{a(i) = 1\}) = \prod_{i=m+1}^{n} P(a(i) = 1)$$

$$= (1 - p)^{n-m} > 0$$

Thus the best deterministic bound one can give is $A(n) - A(m) = n - m$ which means that this Bernoulli flow can not be bounded efficiently by a deterministic arrival curve. To make things worse let the increments be independent exponentially distributed with parameter $\lambda$. Now assume we have a bound $B \geq 0$. With the cumulative distribution function we are able to determine the violation probability of this bound $B$:

$$P(a(i) > B) = 1 - F(B) = e^{-\lambda B} > 0$$

This means, that there is always a chance, that the bound is violated, no matter how large we choose $B$ which implies $a(i) = \infty$ as a deterministic bound and thus the impossibility of this attempt.

In order to solve this problem stochastic arrival curves were introduced, which intuitively bound an incoming flow with a small probability of being invalid, also called the violation probability. In [7] Chang introduced moment generating functions for establishing arrival curves. To understand this concept, we have to define moment generating functions first.

**Definition 1.2.** The moment generating function (MGF) of a real-valued random variable $X$ is defined by:

$$\phi_X : \mathbb{R} \rightarrow \mathbb{R}$$

$$\theta \mapsto \mathbb{E}(e^{\theta X})$$
Now let us formalise the idea of a probabilistic bound stated above. We are interested in a bound with (for all \( n, m \in \mathbb{N} \))

\[
P(A(n) - A(m) > \alpha(n - m)) \leq \epsilon
\]

When applying the well-known inequality of Chernoff\(^1\) on the expression above we get

\[
P(A(n) - A(m) > \alpha(n - m)) \leq e^{-\theta \alpha(n - m)} \cdot \phi_{A(n) - A(m)}(\theta)
\]

In order to establish a probabilistic bounds on the arrivals, one has to provide an upper bound on the MGF of \( A(n) - A(m) \), thus motivating the following definition.

**Definition 1.3.** A flow \( A \) is said to be \((\sigma(\theta), \rho(\theta))\)-bounded for some \( \theta > 0 \) if for all \( 0 \leq m \leq n \) the MGF \( \phi_{A(n) - A(m)}(\theta) \) exists and

\[
\phi_{A(n) - A(m)}(\theta) \leq e^{\theta \rho(\theta)(n - m) + \theta \sigma(\theta)}
\]

holds.

Hence the name \((\sigma, \rho)\)- or MGF-calculus. Basically speaking, these MGF-bounds are the stochastic counterpart of the deterministic arrival curves. In fact, it is not that simple but please keep in mind that this introduction is very limited, for a more in-depth discussion please refer to [7].

Throughout section 2 we will assume the arrivals \( a(i) \) to be independent exponentially distributed with parameter \( \lambda \). Hence, the input flow with \( A(n) = \sum_{i=0}^{n} a(i) \) follows an Erlang(\( \lambda, n \)) distribution:

\[
\phi_A(\theta) = E(e^{\theta A}) = E(e^{\theta(a_1 + \ldots + a_n)}) = E(e^{\theta a_1}) \cdot \ldots \cdot E(e^{\theta a_n})
\]

\[
= ((1 - \frac{\theta}{\lambda})^{-1})^n = (\frac{\lambda}{\lambda - \theta})^n
\]

Thus, in the case of exponentially distributed arrivals (with parameter \( \lambda \)) the MGF can be bounded by \( \rho(\theta) := \frac{1}{\theta} \ln(\frac{\lambda}{\lambda - \theta}) \) and \( \sigma(\theta) := 0 \).

\(^1\)X real-valued r.v., \( x \in \mathbb{R} \) then for all \( \theta > 0 \) it holds \( P(X > x) \leq e^{-\theta x} \phi_X(\theta) \)
Now that arrivals are sufficiently covered, we will investigate the service a node offers to a flow. In general, the concept of (stochastic) service curves is less intuitive than the concept of (stochastic) arrival curves, the focus is to provide a sufficient amount of formalism instead of speaking verbosely about the intuition behind these definitions. For an extensive survey please refer to [12]. Let the doubly indexed process be denoted by $A(m, n) = A(n) - A(m)$ for every $n \geq m \geq 0$.

**Definition 1.4.** Let $\Lambda(N_0) \subset N_0 \times N_0$ be an index set such that $(i, j) \in \Lambda(N_0)$ implies $i, j \in N_0$ and $i \leq j$. A Service is $(\sigma(\theta), \rho(\theta))$-bounded for some $\theta > 0$, if $\phi_{S(m, n)}(-\theta)$ exists and it holds:

$$\forall (m, n) \in \Lambda(N_0) : \phi_{S(m, n)}(-\theta) \leq e^{\theta \rho(\theta)(n-m) + \theta \sigma(\theta)}$$

Please note, that in general it holds $\rho(\theta) \leq 0$.

In the remainder of this work we will assume a constant rate server with rate $c$, hence $S(m, n) := c(n - m)$ for all $(m, n) \in \Lambda(N_0)$, which leads to:

$$\phi_{S(m, n)}(-\theta) = \mathbb{E}(e^{-\theta c(n-m)}) = e^{-\theta c(n-m)}$$

and $\sigma(\theta) := 0, \rho(\theta) := -c$ is a $(\sigma(\theta), \rho(\theta))$-bound for the constant rate server $S$.

In summary the following assumptions are maintained throughout this thesis:

1. Discrete time ($N_0$)

2. A single arriving flow consisting of exponentially $\lambda$ i.i.d. increments, which implies a $(\frac{1}{\theta} \cdot \ln(\frac{\lambda}{\lambda-\theta}), 0)$-boundedness

3. One constant rate server, which is bounded by $(-c, 0)$
2 Theoretical Framework

Throughout this thesis we will assume exponentially distributed arrivals, meaning that the increments $a_i$ of our arrival flow $A = \sum_{i=1}^{n} a_i$ are stochastic independently distributed and follow an exponential distribution with parameter $\lambda$. As stated before, the increment $a_i$ can be interpreted as the amount of data (or whatever is measured) that arrived between the times $i-1$ and $i$.

Although internet traffic and many other real-world examples are known to be heavy-tailed in terms of probability distribution [18] [9], in SNC it is often assumed that arrivals have an exponentially decaying tail as it renders the analysis much less difficult. As this thesis is a first approach to arrival curve estimation in SNC we will begin with the analytically nice case of exponentially distributed arrivals.

First we will introduce an estimator for the distribution parameter $\lambda$ which returns a probabilistic lower bound, or in other terms, a one-sided confidence interval for $\lambda$. Afterwards we will investigate how the uncertainty that arises from this estimation can be merged with the well-known performance bounds of (stochastic) network calculus, namely the backlog and delay bound; the output bound (also known as output characterization) is omitted.

2.1 Estimating the distribution parameter

In this section we will investigate a simple estimator for the distribution parameter $\lambda$. For the theoretical analysis, assume that we measured samples of an arriving flow $A$, composed of increments $a_1, \ldots, a_n$ which are stochastic independent exponentially distributed random variables with parameter $\lambda$. Then we are able to estimate the unknown $\lambda$ by the following method: The idea is to use equalities between different distributions until we are able to estimate our parameter by considering some quantile [25].

As we already know from the introduction, $A$, as a sum of independent exponentially random variables

$$A := \sum_{i=1}^{n} a_i$$
follows an \( \text{Erlang}(n, \lambda) \) distribution. The density of the Erlang distribution is defined as
\[
f(x) = \begin{cases} \frac{\lambda^n x^{n-1}}{(n-1)!} e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}
\]
where \( \lambda \in \mathbb{R} \) and \( n \in \mathbb{N} \) are the parameters. The Erlang distribution is a special case of the gamma distribution, whose density is given by
\[
f(x) = \begin{cases} \frac{1}{b^p \Gamma(p)} x^{p-1} e^{-\frac{x}{b}} & x > 0 \\ 0 & x \leq 0 \end{cases}
\]
with \( b, p \in \mathbb{R}^+ \). For \( p \in \mathbb{N} \) we can conclude with a basic property of the \( \Gamma \)-function [19]:
\[
f(x) = \frac{1}{b^p \Gamma(p)} x^{p-1} e^{-\frac{x}{b}} = \frac{1}{b^p (p-1)!} x^{p-1} e^{-\frac{x}{b}}
\]
which is an Erlang distribution with parameters \( \frac{1}{b} \) and \( p \).

Similarly it can be seen, that the \( \chi^2 \) distribution with \( 2n \) degrees of freedom is a special case of the gamma distribution (with \( b = 2, p = n \)):
\[
\frac{1}{b^p \Gamma(p)} x^{p-1} e^{-\frac{x}{b}} = \frac{x^{n-1}}{2^n \Gamma(n)} e^{-\frac{x}{2}}, \quad x > 0
\] (1)

Considering the flow \( A \) we can conclude
\[
A \sim \text{Gamma} \left( n, \frac{1}{\lambda} \right).
\]

Using basic properties of the gamma distribution we will scale our random variable \( A \) to be \( \chi^2 \) distributed. This distribution is independent of the unknown \( \lambda \) (since it is now part of the scaling factor) which allows us to give an estimate for the parameter. The scaling property states the following:

If \( X \sim \text{Gamma}(k, \theta) \) then for any \( c > 0, cX \sim \text{Gamma}(k, c\theta) \) (2)

Let therefore be \( c > 0 \) a real constant and \( X \sim \text{Gamma}(k, \theta) \). Then the function \( Y = h(X) := cX \) is bijective with inverse \( h^{-1}(Y) = \frac{Y}{c} = X \). Thus it holds for the cumulative distribution function (cdf):
\[
F_Y(y) = F_X(h^{-1}(y)).
\]
Since \( h(X) \) is differentiable with derivation \( \frac{dX}{dY} = \frac{1}{c} \) we obtain:

\[
f_Y(y) = f_X(h^{-1}(y)) \left| \frac{dx}{dy} \right| = \frac{y^{p-1}e^{-\frac{y}{bc}}}{(bc)^p \Gamma(p)} \left| \frac{1}{c} \right|
\]

which is the probability density function of a gamma distribution with parameters \( p \) and \( bc \).

If we now combine these properties we can scale our flow \( A \) to be \( \chi^2 \) distributed and therefore estimate \( \lambda \).

\[
A \sim \text{Gamma} \left( n, \frac{1}{\lambda} \right) \quad \Rightarrow \quad 2 \cdot \lambda \cdot A \sim \text{Gamma}(n, 2) \sim \chi^2(2n)
\]

or equivalently:

\[
2\lambda \sum_{i=1}^{n} a_i \sim \chi^2(2n)
\]

Now, by considering the \( \alpha \)-quantile of the \( \chi^2(2n) \)-distribution, we obtain a value \( x \) with \( \mathbb{P}(X \leq x) = \alpha \) or, equivalently \( \mathbb{P}(x \leq X) = 1 - \alpha \), for any random variable \( X \) with \( X \sim \chi^2(2n) \).

Thus, an one-sided confidence interval for \( \lambda \) with confidence level \( 1 - \alpha \) can be estimated as follows:

\[
\mathbb{P} \left( \chi^2_{\alpha}(2n) \leq 2 \cdot \lambda \cdot A \right) = 1 - \alpha = \mathbb{P} \left( \frac{\chi^2_{\alpha}(2n)}{2A} \leq \lambda \right).
\]

This confidence interval is exact for any number of samples.

For convenience and to underline the semantics we define \( \lambda_{\text{min}} := \frac{\chi^2_{\alpha}(2n)}{2A} \).

In this case it suffices to provide a lower one-sided confidence interval for \( \lambda \) since all presented bounds get worse the smaller the parameter is. This is also what we would expect intuitively: The smaller the parameter of an exponential distribution gets, the larger the variates get (since the probability for large values gets higher).

As a summary of the consideration above we obtain the following theorem:
Theorem 2.1. Let $\lambda \in \mathbb{R}^+$ be a parameter, $A = \sum_{i=0}^{n} a_i$ be a flow and $a_i$ independently exponential distributed random variables with parameter $\lambda$. Then a one-sided confidence interval for $\lambda$ with confidence level $1 - \alpha$ is given by $[\lambda_{\text{min}}, \infty) = \left[ \frac{\chi^2_{\alpha}(2n)}{2A}, \infty \right]$, i.e. it holds

$$P(\lambda \in [\lambda_{\text{min}}, \infty)) = P(\lambda_{\text{min}} \leq \lambda) = 1 - \alpha.$$ 

Please note, that the computation of the confidence interval can be done very efficiently, as the $\alpha$-quantile is constant if $n$ is constant. This can (and will) be achieved by utilizing a sliding window of size $n$ over all the measured samples. Furthermore, if the parameter is estimated at every time step, the sum of increments will differ only by two samples, thus making computation very fast.

2.2 Performance Bounds

In this section we will see, how the parameter estimation and the performance bounds may be combined. They key idea is to merge the uncertainty stemming from the confidence interval with the probability of the backlog bound, which is done by using the law of total probability. Then an upper bound for this merged probability is derived – a stochastic performance bound with estimated parameter. At the end, the derived bound is simplified by using our assumptions of exponential arrivals and a constant rate server.

Before we begin, we need to introduce a short and simple notation for convenience. Let $A_{\psi}$ denote an arrival flow whose increments are exponential($\psi$) i.i.d. distributed.

First, we need to proof two properties for the main theorem, mainly that the relation between parameters of flows can be applied to the flows as well as the backlog and delay:

Lemma 2.2. Let $A_{\lambda}$, $A_{\bar{\lambda}}$ be two arrival flows with $\bar{\lambda} \leq \lambda$ and $N \in \mathbb{N}$ Then it holds:

$$1) \quad P\left( \max_{0 \leq k \leq n} \left\{ A_{\lambda}(n) - A_{\lambda}(k) - S(k,n) \right\} > x \right)$$

$$\leq P\left( \max_{0 \leq k \leq n} \left\{ A_{\bar{\lambda}}(n) - A_{\bar{\lambda}}(k) - S(k,n) \right\} > x \right)$$
\[2) \quad \mathbb{P}\left( \max_{0 \leq k \leq n} \{ A_\lambda(n) - A_\lambda(k) - S(k, n + N) \} > 0 \right) \leq \mathbb{P}\left( \max_{0 \leq k \leq n} \{ A_{\bar{\lambda}}(n) - A_{\bar{\lambda}}(k) - S(k, n + N) \} > 0 \right) \]

which intuitively states, that if one parameter is worse than another, then the corresponding flows and therefore backlog and delay share this relation.

**Proof.** This proof uses the methodologies of probability coupling and inverse method and is therefore quite nice [28].

Let \( \lambda, \bar{\lambda} \in \mathbb{R}^+ \) be two parameters with \( \bar{\lambda} \leq \lambda \) for the flow \( A_\lambda = \sum_{i=0}^{n} a_i \), where \( a_i \) are exponential(\( \lambda \)) i.i.d. The increments \( a_i \) can be generated via the inverse method: From the interval \([0, 1]\) values \( u_i \) are chosen independent and uniformly random. Then, via the inverse cdf, these samples are mapped to the desired distribution.

Please note, that for an exponential(\( \lambda \)) distribution, the inverse cdf is given by

\[ F^{-1}_\lambda(p) = -\frac{\ln(1 - p)}{\lambda} \]

and for every \( p \in [0, 1] \) it holds: \( F^{-1}_\lambda(p) \leq F^{-1}_{\bar{\lambda}}(p) \)

This means, that the flow \( A_\lambda \) may be represented by independent uniformly random distributed variables \( u_i \):

\[ A_\lambda = \sum_{i=0}^{n} F^{-1}_\lambda(u_i) \]

Now we will construct a flow \( A_{\bar{\lambda}} \) which is coupled with \( A_\lambda \). Therefore we will take the same \( u_i \) but apply a different inverse cdf to these random variables:

\[ A_{\bar{\lambda}} := \sum_{i=0}^{n} F^{-1}_{\bar{\lambda}}(u_i) \]

Because \( \bar{\lambda} \leq \lambda \) holds and with the considerations in (2.2) we get \( A_\lambda(n) \leq A_{\bar{\lambda}}(n) \) for every \( n \), see figure 2 for a graphical representation of the procedure.
Thus, we can now conclude the desired property for the coupled random variables:

\[
P\left( \max_{0 \leq k \leq n} \{ A_\lambda(n) - A_\lambda(k) - S(k, n) \} > x \right) \leq P\left( \max_{0 \leq k \leq n} \{ A_\lambda(n) - A_\lambda(k) - S(k, n) \} > x \right) \tag{3}
\]

\[
\leq P\left( \max_{0 \leq k \leq n} \{ A_\lambda(n) - A_\lambda(k) - S(k, n) \} > x \right) \tag{4}
\]

Let \( A_\lambda' \) and \( A_\lambda'' \) be flows (i.e. random variables) with \( \text{exponential}(\lambda) \) and \( \text{exponential}(\bar{\lambda}) \) i.i.d. increments, respectively. Hence, they follow the same distribution as \( A_\lambda \) and \( A_\lambda'' \):

\[
P(A_\lambda \leq x) = P(A_\lambda' \leq x) \quad \text{and} \quad P(A_\lambda'' \leq x) = P(A_\lambda'' \leq x)
\]

for all \( x \).

This allows us to conclude 1) with (3) and since difference and maximum are measurable functions. 2) is deduced with the exact same argumentation.
Figure 2: Illustration of the inverse method and probability coupling, used in lemma 2.2
Finally, we are able to prove the stochastic backlog bound with respect to parameter estimation.

**Theorem 2.3** (Backlog Bound with estimated parameter). Let $A$ be a single flow with exponential($\lambda$) i.i.d. increments arriving at a node serving with constant rate $c$, $k \in \mathbb{R}$, $n \in \mathbb{N}$ arbitrarily chosen and $\lambda_{\text{min}}$ an estimator for $\lambda$ with $\mathbb{P}(\lambda_{\text{min}} \leq \lambda) = 1 - \alpha$. Then the backlog bound with respect to parameter estimation is given by (for all $\theta$ with $0 < \theta < \lambda_{\text{min}}$):

$$
\mathbb{P}(q_{\lambda\mathbf{n}}(n) > k) \leq e^{-\theta k} \cdot \frac{\left(\frac{\lambda_{\text{min}}}{\lambda_{\text{min}} - \theta c} e^{-\theta c}\right)^{n+1} - 1}{\frac{\lambda_{\text{min}}}{\lambda_{\text{min}} - \theta c} e^{-\theta c} - 1} \cdot (1 - \alpha) + \alpha
$$

**Proof.** First, we introduce a new notation: Let $q_\psi$ denote the backlog of a single constant rate server with rate $c$ and an arrival flow, which follows an exponential distribution with parameter $\psi$.

Assume that the increments of a flow $A$ follow an exponential distribution with $\lambda$ and we have some estimator $\lambda_{\text{min}}$ with $\mathbb{P}(\lambda_{\text{min}} \leq \lambda) = 1 - \alpha$. Then, using the law of total probability $\text{2}$, we can partition the probabilities:

$$
\mathbb{P}(q(n) > k) = \mathbb{P}(q_{\lambda\mathbf{n}}(n) > k \mid \lambda_{\text{min}} \leq \lambda) \mathbb{P}(\lambda_{\text{min}} \leq \lambda) \underbrace{= 1 - \alpha}_{= 1 - \alpha} \\
+ \mathbb{P}(q_{\lambda\mathbf{n}}(n) > k \mid \lambda_{\text{min}} > \lambda) \mathbb{P}(\lambda_{\text{min}} > \lambda) \underbrace{= \alpha}_{\leq 1} \\
\leq \mathbb{P}(q_{\lambda\mathbf{n}}(n) > k \mid \lambda_{\text{min}} \leq \lambda)(1 - \alpha) + \alpha
$$

Note, that the events of the conditionally probability in the last equation are to some extent independent: $\lambda_{\text{min}} \leq \lambda$ is a statement on the estimator $\lambda_{\text{min}}$, whereas $q_{\lambda\mathbf{n}}(n) > k$ is a statement related to $\lambda$.

To derive the backlog bound, we have to establish an inequality for the

\(^{2}\mathbb{P}(A) = \sum_{n} \mathbb{P}(A \mid P_n) \mathbb{P}(P_n), \) for any finite or countably infinite disjoint partition of the sample space \(\{P_n \mid n = 1, 2, \ldots\}\) and any event $A$.
conditional probability:

\[
\mathbb{P}(q_{\lambda}(n) > k \mid \lambda_{\text{min}} \leq \lambda)(1 - \alpha) + \alpha
\]

\[
\leq \sup_{\lambda_{\text{min}} \leq \lambda} \mathbb{P}(q_{\lambda}(n) > k \mid \lambda_{\text{min}} \leq \lambda)(1 - \alpha) + \alpha
\]

\[
= \sup_{\lambda_{\text{min}} \leq \lambda} \mathbb{P}(q_{\lambda}(n) > k)(1 - \alpha) + \alpha
\]

\[
\leq \mathbb{P} \left( \max_{0 \leq j \leq n} \{ A_{\lambda}(n) - A_{\lambda}(j) - S(j,n) \} > k \right) (1 - \alpha) + \alpha
\]

\[
\leq \mathbb{P} \left( \max_{0 \leq j \leq n} \{ A_{\lambda_{\text{min}}}(n) - A_{\lambda_{\text{min}}}(j) - S(j,n) \} > k \right) (1 - \alpha) + \alpha
\]

\[
\leq e^{-\theta k + \theta (\sigma_{A_{\lambda_{\text{min}}}}(\theta) + \sigma_{S}(\theta))} \cdot \sum_{i=0}^{n} e^{\theta i \left( \rho_{A_{\lambda_{\text{min}}}}(\theta) + \rho_{S}(\theta) \right) / \lambda_{\text{min}} \lambda_{\text{min}} - \theta}} (1 - \alpha) + \alpha
\]

where “BB” is the abbreviation for the traditional backlog bound.

Please recall, that we assumed a constant rate server (with rate \( c \)) earlier from which we derived \( \sigma_{A}(\theta) = \sigma_{S}(\theta) = 0 \) as well as \( \rho_{A}(\theta) = \frac{1}{\theta} \ln \left( \frac{\lambda}{\lambda - \theta} \right) \) and \( \rho_{S}(\theta) = -c \).

Applying this the expression above simplifies to:

\[
e^{-\theta k} \cdot \sum_{i=0}^{n} e^{\theta i \left( \frac{1}{\theta} \ln \left( \frac{\lambda_{\text{min}}}{\lambda_{\text{min}} - \theta} \right) \right) - c}} (1 - \alpha) + \alpha
\]

\[
= e^{-\theta k} \cdot \sum_{i=0}^{n} \left( \frac{\lambda_{\text{min}}}{\lambda_{\text{min}} - \theta} \right)^i \cdot \left( e^{-\theta c} \right)^i (1 - \alpha) + \alpha
\]

\[
\leq e^{-\theta k} \cdot \sum_{i=0}^{n} \left( \frac{\lambda_{\text{min}}}{\lambda_{\text{min}} - \theta} \cdot e^{-\theta c} \right)^i (1 - \alpha) + \alpha
\]

The remaining sum can be brought into a closed form by utilizing the formula for partial sums of geometric series

\[
\sum_{i=0}^{n} q = \frac{q^{n+1} - 1}{q - 1}
\]
with \( q \in \mathbb{R} \), hence
\[
e^{-\theta k} \cdot \left( \frac{\lambda_{\min}}{\lambda_{\min} - \theta} e^{-\theta c} \right)^{n+1} - 1 \cdot (1 - \alpha) + \alpha
\]
which concludes our proof.

Last but not least, we obtain the delay bound, a very similar result with a very similar proof.

**Theorem 2.4** (Delay Bound with Estimated Parameter). Let \( A \) be a single flow with exponential (\( \lambda \)) i.i.d. increments arriving at a node serving with constant rate \( c \); \( n, N \in \mathbb{N} \) arbitrarily chosen and \( \lambda_{\min} \) an estimator for \( \lambda \) with \( \mathbb{P}(\lambda_{\min} \leq \lambda) = 1 - \alpha \). Then the delay bound with respect to parameter estimation is given by (for all \( \theta \) with \( 0 < \theta < \lambda_{\min} \)):
\[
\mathbb{P}(d(\lambda,n) > N) \leq e^{-\theta c N} \cdot \left( \frac{\lambda_{\min}}{\lambda_{\min} - \theta} e^{-\theta c} \right)^{n+1} - 1 + \alpha
\]

**Proof.** Please keep in mind that it holds \( \sigma_A(\theta) = \sigma_B(\theta) = 0 \) as well as \( \rho_A(\theta) = \frac{1}{\theta} \ln \left( \frac{\lambda}{\lambda - \theta} \right) \) and \( \rho_B(\theta) = -c \).
The abbreviation “DB” refers to the traditional delay bound.
\[
\begin{align*}
\mathbb{P}(d(\lambda,n) > N) & \leq \sup_{\lambda_{\min} \leq \bar{\lambda}} \mathbb{P} \left( \max_{0 \leq k \leq n} \{ A_{\lambda}(k,n) - S(k,n) \} > N \mid \lambda_{\min} \leq \bar{\lambda} \right) \cdot (1 - \alpha) + \alpha \\
& = \mathbb{P} \left( \max_{0 \leq k \leq n} \{ A_{\lambda_{\min}}(k,n) - S(k,n) \} > N \right) \cdot (1 - \alpha) + \alpha \\
& \leq e^{N \rho_S(\theta) + \theta (\sigma_{A,\lambda_{\min}}(\theta) + \sigma_S(\theta))} \cdot \sum_{i=0}^{n} e^{\theta i (\rho_{A,\lambda_{\min}}(\theta) + \rho_S(\theta))} (1 - \alpha) + \alpha \\
& = e^{-\theta c N} \cdot \sum_{i=0}^{n} e^{i \ln \left( \frac{\lambda_{\min}}{\lambda_{\min} - \theta} e^{-\theta c} \right)} \cdot e^{-\theta i c} + \alpha \\
& \leq e^{-\theta c N} \left( \frac{\lambda_{\min}}{\lambda_{\min} - \theta} e^{-\theta c} \right)^{n+1} - 1 (1 - \alpha) + \alpha
\end{align*}
\]
\[\square\]
3 Evaluation

In this section we will evaluate the analytical results from section 2 in a simulated environment. Therefore, Markovian arrival processes (MAP) [21] are introduced and the assumption of identically distributed arrivals will be dropped, which (formally) implies, that the analytical results will not hold anymore. However, in section 3.3 we will see, that under certain assumptions the established bounds still hold empirically.

3.1 Introduction to Markov Chains

Due to the lack of a real-world setup and for reasons of simplicity, a simple simulation framework was set up. In order to randomly generate arrivals with an “interesting” shape, Markov models were used to create four different classes of exponentially distributed arrivals.

First, let us briefly investigate Markov chains [14] and introduce the principle of Markovian arrival processes (MAP).

Definition 3.1 (Markov Chain over a finite set of states and discrete time).
Let $S = s_1, \ldots, s_n$ be a finite set of states and time $t = 0, 1, \ldots$ be discrete. Then the stochastic process $X = (X_t)_{t=0,1,\ldots}$ with values in $S$ is called a Markov Chain on $S$ if the following equality holds:

$$
P(X_{t+1} = s_{j_{t+1}} \mid X_t = s_{j_t}, X_{t-1} = s_{j_{t-1}}, \ldots, X_0 = s_{j_0}) = P(X_{t+1} = s_{j_{t+1}} \mid X_t = s_{j_t})$$

The equality is also called the Markov property which means that, given the present state, the future and past states are independent or equivalently, that the future behaviour of the system only depends on the current state.

Since (in this definition) the state space is finite the transition probabilities can be represented by a transition matrix with

$$p_{i,j}(t) = P(X_{t+1} = j \mid X_t = i), \quad i, j = 1, \ldots, n$$

which can be interpreted as the probability of the system to change into state $j$ at time $t$ given that it is currently in state $i$.

From a computer scientist’s point of view these finite Markov chains can be interpreted as a type of finite automaton with probability driven transitions.
Let us have a look at an example of Markov chains. One of the most prominent (and overused) examples is the weather model. Assume that the weather has only two different states: Sunny and Rainy. Then, given the weather on the preceding day, can we predict the weather for today? The answer is (as one would expect) yes, by using a Markov chain.

The time model is discrete (as we are talking about different days) and the states of the Markov chain correspond to the two weather states, rainy and sunny. Let us assume the following transition probabilities: A sunny day is 70% likely to be followed by another sunny day and a rainy day is 40% likely to be followed by another rainy day. As the sum of a states outgoing edges has to be 1 the following transition matrix can be derived:

\[
\begin{pmatrix}
0.7 & 0.3 \\
0.4 & 0.6
\end{pmatrix}
\]

Or in a nicer graphical form:

Now the weather for today can be predicted by multiplication of a state vector, which represents yesterdays weather conditions, with the transition matrix. Assume that at day 0 the weather was rainy, which implies the following state vector \((0 \ 1)\). Then by multiplication of the two, the weather on day 1 can be predicted:

\[
(0 \ 1) \cdot \begin{pmatrix}
0.7 & 0.3 \\
0.4 & 0.6
\end{pmatrix} = (0.4 \ 0.6)
\]

Hence, there is a 60% chance, that day 1 will be a sunny day. The following days can be estimated similarly by repeated multiplication of the transition matrix with the initial state vector.

This concludes the short introduction to Markov chains, for a more detailed introduction and analysis, please refer to the extensive literature, such as [22] [26] and [27].
3.2 A Markov Modulated Arrival Process

In the discussion so far, we have always adopted a rather simple model for the arriving flow: Some exponential($\lambda$) independent and identically distributed increments. In order to generate arrivals with a more realistic and interesting shape, we will use the concept of Markov modulated arrival processes.

Originally, Markov modulated arrival processes were introduced in the field of queuing theory by [20]. In contrast to the traditional assumption of Poisson distributed arrivals a Markov chain was used in order to generate a more realistic arrival flow which is independent of the Markov process. This traffic model is called Markov modulated Poisson process (MMPP) and is, basically speaking, an $n$-state Markov chain which generates Poisson distributed arrivals with rate $\mu_i$ ($i \in \{1, \ldots, n\}$). Every state can be interpreted as a different traffic class, ranging from light to heavy traffic. At each time step arrivals are generated according to the parameter of the current state.

In our simulation the underlying Markov chain consists of four states, each representing a different class of traffic where state 1 means no arrivals, state 2 light arrival traffic, state 3 medium traffic and state 4 heavy traffic. The model is very similar to a MMPP model but, in contrast to generating Poisson distributed arrivals, exponentially distributed arrivals are returned. This generalization is, among others, called an Markov Modulated (arrival) process (MMP) or Markovian arrival process (MAP). For a detailed introduction and discussion of Markov modulated arrival processes, see chapter 8 of [21].

Please recall that we assumed exponentially distributed arrivals and a constant rate server with rate $c$. The stability conditions states that it must hold $c > \frac{1}{\lambda}$, to achieve a stable system (If this inequality does not hold there will be more arrivals than the server can handle, thus leading to infinite expected backlog).

Deriving from this stability condition we choose the distribution parameter of the 4 traffic classes as follows: $\lambda_{no} := \infty$, $\lambda_{light} := \frac{4}{c}$, $\lambda_{medium} := \frac{2}{c}$ and $\lambda_{heavy} := \frac{5}{4c}$, which implies utilizations of 0%, 25%, 50% and 80%, respectively. Figure 3 illustrates the Markov chain.

In the following section we will discuss issues that arise alongside with the application of a MAP, as well as the preconditions the Markov chain has to fulfil in order to receive reasonable results.
Figure 3: The MAP used in the evaluation. The chain is built to stay in the light and medium traffic states for most of the time.

3.3 Theory vs. Practice

In the analytical considerations we assumed independently, identically exponential($\lambda$) distributed arrivals. Although this assumption is nice in theory, it is, on the one hand, not very realistic and, on the other hand, does not exploit the full potential of the dynamic method presented in this work. When using the Markov modulated arrival process introduced above, the process itself will be more realistic and interesting, but from a formal point of view the confidence interval (and thus the performance bounds) will not hold.

The consideration is as follows: Since Markov chains are random processes, the chain may change its state in every time step. Moreover, since samples are generated randomly, too, we can never be sure whether the measured sample is just an unlikely extreme value or an indicator for a change of $\lambda$. Thus, we cannot estimate $\lambda$ with any fixed confidence at every time step and therefore can not provide any performance bounds.

On a more formal level the omission of the identically distribution assumption leads to invalid proofs, since the sum of $n$ independent exponential($\lambda$) but not identically distributed random variables is not Erlang($n, \lambda$) distributed, which renders the parameter estimator as well as the performance bounds untenable.

However, this result is not as devastating as it may seem: If the arrival Markov chain does not change its state regularly, the performance bounds can be applied to the time with no change in parameter, providing us with reliable and proven results. Unfortunately, as we can never know at which points in time the parameter is changing, we can never be absolutely sure, that the computed bounds really hold. In section 3.3.2 it will be shown, that although no formal guarantees can be given, the estimated bounds still
perform quite well in comparison to the traditional ones.

3.3.1 Evaluation of the Parameter Estimator

In this section some results on the quality of the parameter estimator are presented. These results are related to the application in SNC, thus basic estimator properties as consistency, bias, . . . are not investigated. In the simulation a moving window of fixed size $w$ is laid upon the measured samples and shifted over time. This approach has the advantage, that older samples do not influence the estimation process and, furthermore, is much more practical, since only a fixed amount of data has to be stored.

Please recall the definition of the estimator from section 2

$$P\left(\frac{\chi^2_n(2n)}{2A} \leq \lambda\right) = 1 - \alpha \sim \lambda_{min} := \frac{\chi^2_n(2n)}{2 \cdot A} = \sum_{i=0}^{n} a_i.$$ 

As one can see from the formula above, the estimated parameter $\lambda_{min}$ gets smaller, the higher the confidence level (and thus the smaller $\alpha$) becomes. This is also what is intuitively expected, since $\lambda_{min}$ is a lower bound on $\lambda$.

Let us now investigate the quality of parameter estimation for the Markov modulated arrival process presented above. Figure 5 shows the estimation of a constant parameter $\lambda = 1$, based on 20000 i.i.d. exponential ($\lambda$) samples for different window sizes ($w \in \{50, 100, 500, 1000\}$). As one would expect, the estimation quality gets better (i.e. less jitter), the larger $w$ becomes.

For a more interesting result please consider the sample run of our MAP in figure 6. Again, 20000 samples were generated. The rate parameters were chosen as $\lambda_{light} = 1, \lambda_{med} = \frac{1}{2}$ and $\lambda_{heavy} = \frac{1}{4}$. In the illustration, the black line refers to the actual parameter $\lambda$ and the coloured lines to the estimation. To avoid strange behaviour, the estimator outputs 0 if, and only if, the number of samples is less than $w$.

As one can see the estimator reacts relatively well to changes of $\lambda$, the quality and speed of adaption depending on the window size $w$. Moreover, as it is clearly visible in figure 6, when $\lambda$ changes it takes at most one window size for the estimator to react. Arising thereby are two complementary problems:
The estimation quality gets better the higher $w$ becomes but the time to react to changes becomes longer. However, one should keep in mind, that an unstable estimator leads in some cases to the violation of estimated bounds:

Let $\lambda_i$ denote the parameter the MMAP Markov chain outputs at point in time $i$. If it holds $\lambda_{i-1} < \lambda_i$ and $\lambda_i = \lambda_i + w + x$, $x = 1, 2, \ldots$ ($\lambda$ changes at $i$ and stays constant for at least one window size) then the estimated bound will be sub optimal but still correct, since a smaller parameter is always “worse”. If, however, $\lambda$ becomes smaller (and afterwards stays constant for at least one window size) the estimator will not be correct for at most $w - 1$ points in time. Figure 4 shows an example, where the estimator outputs bogus results due to parameter oscillation.

Thus one has to choose the window size carefully and with respect to the application. This issue is, among others, addressed in the next section.

![Figure 4: An example how too much oscillation in $\lambda$ breaks the estimation procedure ($w = 500, \alpha = 10^{-6}$).](image)
Figure 5: Estimation of a constant parameter with different window sizes with $\alpha = 0.01$. 
Figure 6: Estimation in a MAP setting with different window sizes and \( \alpha = 0.01 \).
3.3.2 Comparison to traditional Performance Bounds

In this section we will compare the derived bounds from section 2 to the traditional performance bounds. Moreover, we will introduce a dynamic point of view for the computation of these bounds. Please recall, that in SNC, incoming traffic is bounded by arrival curves (or more generally speaking: envelopes) which is done using the \((\sigma(\theta), \rho(\theta))\) mechanism. This provides a bound to the MGF of the corresponding arrival flow and is the only knowledge that remains on the arriving flow. Moreover, SNC identifies the starting point of the system to be analysed with the time point 0 and derives bounds relatively to this starting point. Thus one could say, SNC has a static point of view on the system: Arrivals and Services have to be bounded beforehand and every point in time is analysed relatively to 0 therefore including the whole history of the system until that point in time in the computation.

This behaviour is perfectly fine and legitimate for a static analysis. However, for a dynamic analysis of a running system, this approach is not very useful, since benefits of the dynamic information cannot be utilized. This gap is filled by the theory we developed in section 2, although one must keep in mind, that it only provides formal guarantees for exponential i.i.d. arrivals. By utilizing the parameter estimator and by only considering the last \(w \in \mathbb{N}\) samples (or equivalently: by regarding a moving window of size \(w\) on all samples) we are enabled to compute a dynamic view on the arrival process.

Albeit, a dynamic view on the arrivals and a (still) static view on time leads to unwanted behaviour: Consider a system with low utilization for some time until a short burst occurs, leading to a small amount of backlog. In terms of exponential parameters \((\lambda)\) this implies a short drop \((\bar{\lambda})\), which will be noticed by the estimator, thus leading to a short drop in the estimated parameter. If \(\bar{\lambda}\) is used to calculate the expected backlog at the time of the burst, it will lead to surprisingly bad results. Now the crux here is, that the view on time is still static, leading SNC to the assumption that the arrivals have always been \(\lambda\)-distributed.

To circumvent this mixture of viewpoints, in practice we re-set the 0 point in time to the current time, in particular to the point in time the last sample was measured.

However, since in (S)NC there is no delay and no backlog at time step 0, one has to keep in mind that by re-setting the origin of time, the system may
be already backlogged and delayed. Fortunately, as we will see in Theorem 3.1, the existing theories still hold: it suffices to simply add existing backlog and delay to the current consideration.

Theorem 3.1. Let $S$ be a constant rate server and $A_\lambda$ an arriving flow at $S$. Then for the same system but without initial backlog/delay and every point in time $n$ and for every $B, C \in \mathbb{R}$, $M, N \in \mathbb{N}$ it holds:

1. $P(q(n) > B + C \mid q(0) = C) \leq P(q'(n) > B)$

2. $P(d(n) > N + M \mid d(0) = M) \leq P(d'(n) > N)$

Where $q'$ and $d'$ belong to a server $S'$ identical to $S$, but without the initial backlog/delay. The equation states, that any existing backlog $C$ and/or delay $M$ can be ignored in the calculations.

Proof. ad 1)

Let $\bar{X}$ be a backlogged system (containing a constant rate server $S$ and an arriving flow $A$) with $C$ backlog, i.e. $q_{\bar{X}}(0) = C$ and $X$ be the exact same system, without the backlog. Then, similar to the methodology of stochastic coupling, $X$ will “mimic” every action of $\bar{X}$. This means, that if at a point in time $i$ (with arbitrary $i \in \mathbb{N}$) the backlog at $\bar{X}$ increases, it will increase in the same amount at $X$, which implies $q_{\bar{X}}(i) = q_X(i) + C$. However, if the backlog decreases at time point $i$, it holds $q_{\bar{X}}(i) \leq q_X + C$ because the possible amount of backlog is lower bounded by 0. Thus, we can conclude the following:

$$P(q_{\bar{X}}(i) > B + C \mid q_{\bar{X}}(0) = C) \leq P(q_X(i) + C > B + C \mid q_{\bar{X}}(0) = C)$$

Please note, that in the second conditional probability, the two events are independent, hence it can be simplified to

$$P(q_X(i) + C > B + C) = P(q_X(i) > B)$$

which concludes our proof, since $X$ behaves like $\bar{X}$ but without the initial backlog.
**Definition 3.2 (”Inverse” Backlog and Delay).** Given a violation probability $p \in [0, 1]$ and a point in time $n \in \mathbb{N}$. Then the inverse backlog and delay are defined respectively:

1. $q_p^{-1}(n) := \min\{B \mid \mathbb{P}(q(n) > B) \leq p\}$
2. $d_p^{-1}(n) := \min\{N \mid \mathbb{P}(d(n) > N) \leq p\}$

Now we are finally able to evaluate the results from section 2 and give a comparison between the traditional static and our proposed dynamic approaches. Please note that due to comparability reasons, the $(\sigma(\theta), \rho(\theta))$-bound of the static approach is not tight when dealing with changing parameters. We chose the “bounding” parameter as follows: $\lambda := \min\{\lambda_1, \ldots, \lambda_n\}$ with $\lambda_i$ being the parameter used in the $i$-th state of the Markov chain – in other words the minimal parameter that is used in the MAP. In the plots the inverse bounds are rounded down to the next integer to suppress jitter. Moreover, when computing dynamic bounds, one has to take in account the existing backlog/delay in the system which is in this case added to the estimated inverse bound. In order to keep the already large number of plots to a minimum, mostly backlog was regarded in the considerations, as the behaviour for delay is very similar, see figure 8.

The implementation was done in R [24], since it is nice for plots and high level mathematical operations, although tends to be quite tedious sometimes. The performance bounds are computed in a straightforward manner: Depending on whether the bound is dynamic or static, the desired expression is minimized with respect to $\theta$. Regarding the inverse bounds, any existing backlog and delay are added to the estimated bounds, as already suggested by theorem 3.1.

If the number of samples observed so far is smaller than the window size, the estimated inverse bound will output 0. Moreover, if the future window is
larger than 1, some points in time (between $w + 1$ and $w + f$, in particular) are evaluated from time point $w + 1$, which explains the short plateau at the beginning of the changing parameter plots.

First, let us investigate a simple setting were the arrivals are i.i.d. with constant parameter $\lambda$. In this scenario the traditional performance bounds are also the best, since the $(\sigma(\theta), \rho(\theta))$-bound on the arriving flow is tight. Please note, that the higher the utilization gets, the longer it takes for the backlog/delay bound to converge. For the following plots we chose a utilization of 50%, in particular $c := 2, \lambda := 1$. We will consider the topic of utilization again later on.

In general it holds, that for growing confidence level (diminishing $\alpha$) the estimated bound gets worse, i.e. it is shifted vertically. Hence, we chose a fixed $\alpha := 10^{-6}$, since a change in confidence level does not yield very interesting results. Similarly, we chose the violation probability $p \in \{10^{-2}, 10^{-4}\}$ – relatively high probabilities in contrast to values used in related work. This is due to the fact, that lower probabilities ($\leq 10^{-6}$) do not change the relationship between estimated and traditional bound but decrease readability of the plots.

The consideration above already indicates, that the crucial parameters are the window size $w$ and how much the point of estimation lies in the future, $f$. Please recall, that in the dynamic model the absolute zero time point is redefined at every time step.

Similar to the evaluation of the parameter estimator, a constant rate server was simulated for a time of 20000 steps and the window size $w$ lies in $\{100, 500, 1000\}$. Regarding the size of the future window $f$, the best bound can be achieved by only estimating the next point in time, in other words $f = 1$. Furthermore, $f$ was chosen as 50% and 100% of the window size, respectively.

As it is clearly visible in figures 9, 11 and 13 the larger $w$ becomes, the more the estimated bounds can thoroughly compete with the traditional performance bounds, for $f = 1$. However, as the future window $f$ grows, the bounds get worse as the parameter directly influences the exponent in the backlog/delay bound, see figures 10, 12 and 14. Albeit, for low utilization, the traditional backlog/delay bound converge fast, thus an even larger future window (e.g. the transition from 50% to 100% of $w$) would not yield such a notable effect.
Figure 7: Change of parameter $\lambda$ over time in the regarded sample run of the MAP. As it is clearly visible, the parameter stays mostly constant – a vital condition for reasonable estimation.
Figure 8: Different Delay Bounds, with $w = 500$ and a utilization of 50% (constant $\lambda = 1$, $c = 2$. It can be seen, that the overall behaviour of the bounds is very similar to the extensively covered backlog bounds.
Figure 9: Backlog bounds of a system with constant parameter, $w = 100$ and $f = 1$. In comparison to larger window sizes, the quality of the bounds is relatively bad and made even worse by the fact that there is considerable jitter.
Figure 10: Backlog bounds of a system with constant parameter, $w = 100$ and $f = 50$. As the future window increases, the negative effects of jitter and $\lambda_{min}$ being a lower bound on $\lambda$ get augmented.
Figure 11: Backlog bounds of a system with constant parameter, \( w = 500 \) and \( f = 1 \). It is clearly visible that the estimated bounds can thoroughly compete with the traditional ones.
Figure 12: Backlog bounds of a system with constant parameter, $w = 500$ and $f = 250$. Although the future window is significantly larger than in figure 10, the larger window size compensates the negative effects to some extent.
Figure 13: Backlog bounds of a system with constant parameter, $w = 1000$ and $f = 1$. As the window size increases the estimation quality becomes better, especially compared to figure 9.
Figure 14: Backlog bounds of a system with constant parameter, $w = 1000$ and $f = 500$. As it can be seen, the better estimation quality due to increased window size does not outweigh the negative effects arising from an increased future window like in figure 12.
With a utilization of 50% everything is fun and games, with decreasing λ things tend to become more ugly. There are two major problems arising as a result from high utilization: First, the fact that \( \lambda_{\text{min}} \) as a lower bound on λ may violate the stability condition (meaning \( \frac{1}{\lambda_{\text{min}}} > c \)), which implies arbitrary large values for inverse backlog, as the bound will not converge. Second, small changes in \( \lambda_{\text{min}} \) as well as \( f \) are augmented which results in astronomic bounds. This can be nicely seen in figure 15, where \( w = 1000, c = 2 \) and \( \lambda = \frac{5}{8} \) which implies a utilization \( u = \frac{1}{2} \cdot \frac{8}{5} = 0.8 \). It may arise the question, why the best estimated bound is significantly better than the traditional. The answer is simple: For high utilization the traditional bound needs some time to converge, this is also nicely visible in the figure. As in the best bound only the next time step (i.e. \( f = 1 \)) is evaluated, the estimated bound benefits from the fact, that the traditional backlog bound has not converged yet.

Last but not least, let us investigate a sample run of the MAP presented above. As discussed before, the presented bounds do not provide any formal guarantee, although it is clearly visible from figures 16, 17 and 18, that these bounds actually hold most of the time. Naturally, the parameter returned by the MAP is not subject to strong oscillation, as indicated in figure 7.

The first striking feature of all plots is, that the best possible bound (\( f = 1 \)) is fairly accurate. This is due to the fact, that between two adjacent time steps, the backlog is not subject to severe changes. Moreover, the estimated bounds are sufficiently loose which leads to the odd situation, that the existing backlog and the estimated backlog are almost always larger than the change of backlog between two time steps: \( q(0) + q_p^{-1}(1) > q(1) \).

In general, as long as the utilization is low, the estimated bound outperforms the traditional bound. However, this relationship is reversed if heavy traffic arrives at the server, clearly visible in figures 16, 17 and 18. This effect is intensified the more the future window \( f \) increases, similar to the discussion before.

Regarding the choice of window size \( w \) one has to balance between small window sizes, implying high jitter (which is also augmented if the utilization is high, see figure 16) and relatively bad bounds compared to large window sizes, resulting in better bounds but longer reaction times of the estimator and thus more possible violations of the estimated bound, see figure 19.
Figure 15: Backlog bounds of a system with high (80%) utilization, with $\lambda = 0.625$, $w = 1000$, $p = 0.01$, $f = 1$ (best) and $f = 1000$ (100%), respectively. The best bound is better than the traditional bound because in this scenario there is a substantial difference between converged and initial backlog bound.
Figure 16: Backlog bounds of a MAP system with changing parameter, $w = 100$. As described in the text, the estimation result is satisfying for low as well as medium utilizations and skyrockets if there is heavy traffic – similar to figures 17 and 18.
Figure 17: Backlog bounds of a MAP system with changing parameter, $w = 500$. Similar behaviour in figures 16 and 18.
Figure 18: Backlog bounds of a MAP system with changing parameter, $w = 1000$. Similar to figures 16 and 17.
Figure 19: Backlog bounds of a MAP system with changing parameter, $w = 1000$. Here, different future window sizes are compared. Moreover, it is clearly visible, that the estimator reacts slowly to changes, due to the large window size. This effect is amplified in combination with large $f$. 
3.3.3 A backlog bound over finite Time Horizons

In this section we will take a short glimpse at the calculation of a backlog bound over finite time horizons. Mainly, we will combine the techniques and results presented in [3] with our dynamic approach we have dealt with so far. Please note, that we will not discuss Extreme Value Theory but use the simple inequality which is directly embedded into SNC.

All the considerations on performance bounds so far only apply to an arbitrary, but fixed point in time. In particular, the backlog bound (and delay bound, respectively) provides guarantees only for single time points. However, there are situations, especially when looking from a more network-engineering point of view, where one is interested in the maximum backlog over a certain time interval. Also in the dynamic approach we so far, there is a considerable interest in backlog bounds over finite time horizons. For example, consider a server with some arrivals which runs a prediction software based on the techniques presented in this thesis. Then, it surely would be nice if the software produces outputs in the form of "In the next \( N \) time steps, the maximum backlog will be \( q \), with probability \( p \)" , thus justifying the need for a finite time horizon backlog bound and linking back to the motivation section of this work.

Assume we are interested in a backlog bound over a finite time horizon of size \( N \). Then we are able to bound the maximum backlog, using the following simple and sub optimal inequality:

\[
P(\max_{0 \leq n \leq N} q(n) > B) = \sum_{n=0}^{N} P(q(n) > B) \leq 2 \sum_{n=0}^{N} P(q(n) > B)
\]  

(5)

From section 6 in [3] we obtain the following optimization problem:

\[
P(\max_{0 \leq n \leq N} q(n) > B) \leq \min_{0 < \theta < \lambda} \sum_{i=0}^{N} e^{-\theta B} \left( \frac{\lambda-\theta e^{-\theta c}}{\lambda-\theta} \right)^{i+1} - 1
\]

Let us now investigate a simple example, similar to the constant parameter scenario presented in 3.3: A server with constant rate \( c \) and i.i.d.
exponentially \( \lambda \) distributed arrivals. Now instead of estimating a single point backlog bound, a finite time interval will be considered. Therefore \( w \) is again in \( \{100, 500, 1000\} \), \( \alpha = 10^{-12} \) and \( p = 10^{-9} \). Moreover, in order to obtain comparable plots to the ones presented by Beck and Schmitt [3] 100 different scenarios with utilization between 0\% and 100\% were considered therefore the mean of 10000 estimations of \( \lambda \) was used in the estimated backlog bound. The results are illustrated in figure 20.

Basically, there are no new insights available from the plot: For low and medium utilizations, the estimated bound is (if \( w \) is sufficiently large) comparable to the traditional bound. However, the heavier the arriving traffic becomes, the worse the estimated backlog bound get. Additionally, when considering sample runs of the MAP, there no new insights as well: The estimated bounds get even worse, due to the suboptimality of the sample path backlog bound (5).

As a summary it can be said, that with simple methods of MGF-calculus it is possible to provide a finite sample path backlog bound with estimated arrivals, although the results are not very convincing.
Figure 20: Comparison of backlog bounds over finite time horizons, where $N \in \{10, 40\}$ The higher the utilization becomes, the less competitive the estimated bounds become.
4 Related and Future Work

4.1 Related Work

This section will briefly cover the related work of this thesis.

As already hinted in the introduction, there are different branches of (S)NC: the MGF-calculus by Chang [7], applied and extended in this work is probably the most famous stochastic expansion of deterministic network calculus – at least according to Google scholar. The MGF-calculus was later on specified by Fidler in [11]. Additionally, there is the statistical calculus by Liebeherr et al. [4] and the stochastic network calculus developed by Jiang and Liu, nicely covered in their book [16].

The assets and drawbacks of the different versions are covered by Fidler in [12], which can also be considered an introduction to service curves. Recently, Ciucu and Schmitt published an article on a misconception in the SNC by Jiang [8], indicating that this SNC is unable to account for statistical multiplexing gain. Moreover, the article provides a nice overview and résumé on the topics related to network calculus.

In chapter 3.3 a MAP was used in order to create arrivals with an more interesting shape. For the evaluation as well as comparison between estimated and traditional bounds, we chose the smallest parameter used in the MAP as lower bound for the arriving flow, which is far from optimal. There has been work by Chang on MAP, in particular simple On-Off models, in [6]. However, only a maximum envelope rate (MER) bound is presented and there is no constructive proof on transforming MER-bounds into $(\rho, \sigma)$. Unfortunately, it seems there exists no work on MAP arrival flow $(\sigma, \rho)$-bounds in a discrete time model.

Finally, we cover additional approaches to parameter estimation in exponential distributions. In this thesis we unrolled and used the estimator presented in [25], which has the nice properties of being an interval estimator alongside of returning exact values for any number of samples. Naturally, there are numerous methodologies for exponential parameter estimation, such as the well-known maximum-likelihood and Bayesian inference point estimators [10], [5] or even multivariate estimators [1].
4.2 Future Work

In this section we will investigate possible areas of future work on topic related to the results presented in this thesis.

While working with and on SNC, especially when performance bounds are to be evaluated in simulations or experiments, the question of units arises: What is the measure for the arriving and departing flow, time, backlog, delay, . . . , how does the choice of units affect the results, i.e. should the measure be small (e.g. byte, milliseconds) or big? To the author’s best knowledge there has not been any published consideration on this issue. Furthermore, some brief simulations indicate, that measures are better to be chosen coarse-grained in order to achieve tighter bounds.

In section 3 a MAP was introduced to generate more interesting arrival flows. This, however, led to the loss of formal guarantees, as the assumption of an identically distributed flow was violated. This assumption is vital for the estimator and the exact \((\sigma(\theta), \rho(\theta))\)-bound, albeit there should exist methodologies in advanced statistics that are able to cope with these more complex scenarios, [23] and [2] seem promising. Thus, another possible scope of future work would be to extend the results presented in this thesis in order to manage scenarios without the identically distribution assumption.

In the consideration so far, only the most simple topology of a single constant rate server as well as one arriving and departing flow was contemplated. Albeit, real-world networks tend to be more complex than this single scenario. Consequently, one could investigate more complex service curves, multiplexing, leftover service, tandem networks, etc. with respect to parameter estimation. Moreover, for the single server and single arrival flow case, the obtained results from section 2 should be easily generalizable from exponential distributions to parametric distributions in general.

Last but not least, another possible direction for future work is an implementation and real-world evaluation of the results. Therefore, one has to investigate fast algorithms for bound computation, which, due to the inherent nature of MGF-calculus, touches the field of optimization. Furthermore, when applying SNC on a running system, the question of cost effectiveness has to be considered: Does the benefit from SNC outweigh the computational costs?
It can be summarized, that along with this thesis a variety of possible questions and directions for future work arise, mainly concerned with the generalization of the results to more complex scenarios.
5 Conclusion

In this thesis we investigated a new approach for arrival curve estimation with basic statistics in a scenario containing a single constant rate server and exponentially i.i.d. arrivals. The motivation behind this work stems from the assumption of complete knowledge in SNC: For computing performance bounds one needs accurate information on the network topology, as well as the structure of service and arriving flows. Whereas the topology and service of real-world scenarios is often known, the structure of arriving flows however is not. The work presented in this thesis can be considered a first solution for this problem. Moreover, a new dynamic point of view was introduced which enables one to apply SNC on running systems and therefore exploit the additional available information of the dynamic behaviour.

The heart of this work lies in the theorems of section 2, where the uncertainty of the proposed parameter estimation is incorporated with the well-known probabilistic performance bounds, namely backlog and delay. In section 3 these two major theorems as well as the parameter estimator were evaluated in a wide range of simulations. In direct comparison to the best traditional bounds, meaning that the parameter of the underlying arrival distribution remains constant, the new bounds are worse than the original bounds but can thoroughly compete for large window sizes. However, as the assumption of identically distributed arrivals was dropped, the estimated bounds outperformed the traditional as long as utilization was low, although all established formal guarantees from chapter 2 were lost. Albeit, for high utilization this relationship is reversed. Furthermore, it was shown that the estimated backlog bound can be applied to provide an upper bound over a finite time horizon and matches up to the traditional bounds.

In summary it can be said, that this thesis has laid the groundwork for a new rewarding and promising area of research on SNC.
References


