# Traffic Modeling with Phase-Type Distributions and VARMA Processes

**Jan Kriege, Peter Buchholz**
Department of Computer Science,
TU Dortmund
{jan.kriege,peter.buchholz}@tu-dortmund.de

Author prepared version of a paper published in

Proc. of the 13th International Conference on Quantitative Evaluation of SysTems (QEST 2016), Springer, 2016.

The original publication is available at www.springerlink.com

http://dx.doi.org/10.1007/978-3-319-43425-4_20

# Traffic Modeling with Phase-Type Distributions and VARMA Processes

Jan Kriege and Peter Buchholz
Department of Computer Science, TU Dortmund
{jan.kriege,peter.buchholz}@tu-dortmund.de

**Abstract**

In discrete event systems event times are often correlated among events of one event stream and also between different event streams. If this correlation is neglected, then resulting simulation models do not describe the real behavior in a sufficiently accurate way. In most input modeling approaches, no correlation or at most the autocorrelation of one event stream is considered, correlation between different event streams is usually neglected. In this paper we present an approach to combine multi-dimensional time series and acyclic phase type distributions as a general model for event streams in discrete event simulation models. The paper presents the basic model and methods to determine its parameters from measured traces.
**Keywords:** input modeling, stochastic simulation, phase type distributions, time series, cross-correlation

## 1 Introduction

The adequate modeling of uncertainty is a key aspect in building realistic simulation models of real systems. The term input modeling subsumes techniques to build a mathematical model for the occurrence of events in a discrete event model [1]. In simulation historically events are characterized by distributions that are selected from a set of standard distributions like exponential, log-normal, Weibull to mention only a few examples [19]. In the recent decade also phase type distributions (PHDs), which originally have been applied in models that are solved analytically or numerically, became more popular in simulation [6, 25]. PHDs allow a detailed approximation of measured densities but the parameter estimation is complex although nowadays algorithms are available that allow a fairly efficient and sufficiently exact determination of distribution parameters.

The use of distributions to model times between the occurrence of events, i.e., the inter-event times, is well established but implicitly assumes identically and independently distributed times. However, in practice, inter-event times are often correlated and the negligence of correlation results in unrealistic models and wrong results [9]. Correlations occur in various application areas like job-processing in manufacturing systems, traffic processes in computer networks or failure processes in technical systems. For modeling such dependencies time series [4] and Markovian arrival processes (MAPs) [23] are used. Classical time series are based on the normal distribution which is often not adequate to describe inter-event times in simulations. MAPs are a natural extension of PHDs but the adequate fitting of parameters to capture correlations is a complex and only partially solved problem such that MAPs often fail if larger correlations over various lags are present. In

[7] the autoregressive to anything (ARTA) approach has been proposed which allows one to combine time series with other distributions where the distribution function can be easily inverted. This approach does not cover PHDs directly and has been extended in [17] to include a sufficiently general subclass of PHDs for modeling the distribution which is then combined with classical time series models like moving average (MA), autoregressive (AR) and autoregressive moving average (ARMA). The corresponding model is sufficiently general to model correlated inter-event times and parameter fitting can be performed efficiently.

In several applications not only the inter-event times of one event type are correlated, also inter-event times of different events show so called cross-correlation. Examples for such cross-correlations can be found in manufacturing systems where different service times are correlated [27], in internet traffic where traffic streams from different sources are correlated [26] or in grids where failure times are correlated in time and space [14]. To model such phenomena vector autoregressive to anything processes (VARTA) have been proposed in [2]. VARTA processes are based on marginal distributions of the Johnson type which allow one to fit the parameters according to the first four moments [19]. Distributions of the Johnson type are not as flexible as PHDs are.

In this paper we extend the approach from [17] to vector autoregressive processes and include also discrete time PHDs. The paper is structured as follows. In the next section we introduce the basic concepts, namely multivariate traces, PHDs, ARMA processes and the VARTA approach. Afterwards, in Sect. 3 vector correlated acyclic phase type processes are defined and the corresponding algorithms for parameter fitting are introduced. Sect. 4 contains some experimental results. The paper ends with the conclusions.

# 2 Background and Definitions

The notation we use in this paper is based on the common notation in the literature on time series [4] which is also adopted in the papers introducing the ARTA and VARTA approach [2]. Vectors and matrices are denoted by bold letters and elements are accessed by putting indices in brackets. If $\boldsymbol{\varepsilon}$ is a column vector, then $\boldsymbol{\varepsilon}'$ is the transposed row vector, $\mathbf{1}$ is the column vector where all elements are $1$.

## 2.1 Multivariate Traces

Since we are interested in modeling several interrelated measurements simultaneously, we assume that these measurements have been recorded in form of a multivariate trace. A $k$-variate trace $\mathcal{T}^{(k)}$ describes $k$ sequences of measurements (i.e. inter-event times, packet sizes etc.). We assume that the elements at position $j$ from all sequences are associated, i.e. they are for example the inter-arrival and service time of a customer. This implies that all sequences are of equal length $r$. We denote the $j$-th entry of the $i$-th sequence by $t_j^{(i)}$. Statistical properties can be computed for a single sequence and the complete trace, e.g. the $j$-th moment of sequence $i$ and the variance are estimated from

$$\hat{\mu}_j^{(i)} = \frac{1}{r}\sum_{m=1}^{r}(t_m^{(i)})^j \quad \text{and} \quad \hat{\sigma}_{(i)}^2 = \frac{1}{r-1}\sum_{m=1}^{r}(t_m^{(i)} - \hat{\mu}_1^{(i)})^2,$$

respectively. If we consider the complete trace the dependencies between the elements are of special interest. For a multivariate trace we can compute the autocorrelations between elements of a single sequence but also correlations between elements of different sequences resulting in correlation matrices $\hat{\boldsymbol{\rho}}_h$ that contain at position $(i_1, i_2)$ the cross-correlation between elements of sequences $i_1$ and $i_2$ that are lag $h$ apart. Element $\hat{\boldsymbol{\rho}}_h(i_1, i_2)$ is estimated as

$$\hat{\boldsymbol{\rho}}_h(i_1, i_2) = \frac{1}{(r-h-1)\hat{\sigma}_{(i_1)}\hat{\sigma}_{(i_2)}} \sum_{m=1}^{r-h} (t_m^{(i_1)} - \hat{\mu}_1^{(i_1)})(t_{m+h}^{(i_2)} - \hat{\mu}_1^{(i_2)}). \tag{1}$$

In input modeling one is interested in fitting these traces, i.e. in estimating the parameters of a distribution or a process such that it resembles the characteristics of the trace. In the following we present some distributions and processes used for input modeling that are relevant for our work.

## 2.2 Phase-Type Distributions

PHDs describe independent and identically distributed random variables as absorption times of a finite Markov chain [23]. PHDs can be defined in continuous (CPHDs) and discrete (DPHDs) time. Most of the existing literature covers CPHDs which will be introduced first. A CPHD of order $n$ consists of $n$ transient and one absorbing state and is defined by an $n \times n$ matrix $\boldsymbol{D_0}$ and an initial distribution vector $\boldsymbol{\pi}$. Matrix $\boldsymbol{D}_0$ is the subgenerator of an absorbing continuous time Markov chain that contains the transition rates between transient states. It holds that $\boldsymbol{\pi}\mathbf{1} = 1, \boldsymbol{D}_0(i,i) < 0, \boldsymbol{D}_0(i,j) \geq 0, i \neq j$ and $\boldsymbol{D}_0\mathbf{1} \leq \mathbf{0}$. Events are generated whenever the absorbing state is reached and the process is restarted immediately afterwards as defined by $\boldsymbol{\pi}$. Properties of the distribution can be defined in terms of this matrix and vector, e.g. for the moments and cumulative distribution function we have

$$\mu_i = E(X^i) = i!\boldsymbol{\pi}\boldsymbol{M}^i\mathbf{1} \quad \text{and} \quad F(t) = 1 - \boldsymbol{\pi} exp(\boldsymbol{D_0}t)\mathbf{1}, \tag{2}$$

respectively, where $\boldsymbol{M} = -(\boldsymbol{D_0})^{-1}$.

DPHDs [3] are defined in a similar way. They are described by an $n \times n$ matrix $\boldsymbol{T}$ that contains the transition probabilities between transient states and an initial distribution vector $\boldsymbol{\tau}$ where $\boldsymbol{T}(i,j) \geq 0, \boldsymbol{T}\mathbf{1} \leq \mathbf{1}$ and for $\boldsymbol{\tau}$ the same constraints as for $\boldsymbol{\pi}$ apply. Then, factorial moments and distribution function are given by

$$m_i = i!\boldsymbol{\tau}(\boldsymbol{I} - \boldsymbol{T})^{-i}\boldsymbol{T}^{i-1}\mathbf{1} \quad \text{and} \quad F(t) = 1 - \boldsymbol{\tau}\boldsymbol{T}^t\mathbf{1}, t \geq 0. \tag{3}$$

In general, it is difficult to obtain the moments $\mu_i$ from the factorial moments $m_i$. However, for the following ideas we only need mean and variance, which are given by

$$\mu_1 = m_1 \quad \text{and} \quad \sigma^2 = 2\boldsymbol{\tau}(\boldsymbol{I} - \boldsymbol{T})^{-2}\boldsymbol{T}\mathbf{1} + \boldsymbol{\tau}(\boldsymbol{I} - \boldsymbol{T}^{-1})\mathbf{1} - (\boldsymbol{\tau}(\boldsymbol{I} - \boldsymbol{T})^{-1}\mathbf{1})^2. \tag{4}$$

Depending on the structure of matrix $\boldsymbol{D_0}$ (or $\boldsymbol{T}$) and vector $\boldsymbol{\pi}$ (or $\boldsymbol{\tau}$) several subclasses of PHDs have been defined in the past. If the states can be ordered such that the matrix becomes upper triangular we have an acyclic CPHD or DPHD. Simpler and well known subclasses include the Exponential, Erlang and Hyper-Erlang distributions in the continuous case and the geometric and negative binomial distributions in the discrete case.

There exists a wide theory on fitting CPHDs to the empirical distribution of a trace. Many approaches are Expectation Maximization (EM) algorithms, which try to maximize the likelihood and work on the complete trace. Since EM algorithms are very slow in the general case, they have been tailored for fitting special subclasses of CPHDs like Hyper-Exponential distributions in [16] or Hyper-Erlang distributions in [28]. Faster approaches usually derive some characteristics like moments from the trace and fit the PHD according to these characteristics. An example for fitting PHDs according to empirical moments can be found in [5]. For DPHDs [3] describes a parameter estimation algorithm. A more complete list on fitting approaches can be found in [6].

## 2.3 (Vector) Autoregressive Moving Average Processes

Autoregressive Moving Average Processes ($ARMA(p,q)$) are well established in time series modeling (see [4]) and are defined as

$$Z_t = \alpha_1 Z_{t-1} + \alpha_2 Z_{t-2} + \ldots + \alpha_p Z_{t-p} + \beta_1 \epsilon_{t-1} + \beta_2 \epsilon_{t-2} + \ldots + \beta_q \epsilon_{t-q} + \epsilon_t \tag{5}$$

where the $\alpha_i$ are autoregressive coefficients, the $\beta_i$ are moving average coefficients and the $\epsilon_t$ are called innovations that have normal distribution with mean zero and variance $\sigma_\epsilon^2$. If the moving average terms in Eq. 5 are omitted the process becomes an $AR(p)$ and if the autoregressive terms are omitted an $MA(q)$.

A generalization of ARMA processes are Vector ARMA processes ($VARMA(p,q)$) that can model multivariate time series [21]. If we assume that the $Z_t$ in Eq. 5 are vectors and the coefficients are defined by matrices we immediately obtain the definition of a $VARMA(p,q)$ process:

$$\boldsymbol{Z}_t = \boldsymbol{\alpha}_1 \boldsymbol{Z}_{t-1} + \boldsymbol{\alpha}_2 \boldsymbol{Z}_{t-2} + \ldots + \boldsymbol{\alpha}_p \boldsymbol{Z}_{t-p} + \boldsymbol{\beta_1 \epsilon_{t-1}} + \boldsymbol{\beta_2 \epsilon_{t-2}} + \ldots + \boldsymbol{\beta_q \epsilon_{t-q}} + \boldsymbol{\epsilon_t} \tag{6}$$

where the $\boldsymbol{Z}_t = (Z_t^{(1)}, Z_t^{(2)}, \ldots, Z_t^{(k)})'$ are $(k \times 1)$ vectors with the observations at time $t$, the $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ are $(k \times k)$ matrices with autoregressive and moving average coefficients, respectively, and $\boldsymbol{\epsilon_t} = (\epsilon_t^{(1)}, \epsilon_t^{(2)}, \ldots, \epsilon_t^{(k)})'$ is a $(k \times 1)$ vector with innovations introducing randomness into the sequence. We will assume that the $\boldsymbol{\epsilon_t}$ are Gaussian with covariance matrix $\boldsymbol{\Sigma}_\epsilon$ and $E[\boldsymbol{\epsilon_t}] = 0$, $E[\boldsymbol{\epsilon_t} \boldsymbol{\epsilon_t}'] = \boldsymbol{\Sigma}_\epsilon$, $E[\boldsymbol{\epsilon_t} \boldsymbol{\epsilon_s}'] = \boldsymbol{0}, s \neq t$. Similar to the univariate case we obtain a $VAR(p)$ for $q = 0$ and a $VMA(q)$ for $p = 0$.

## 2.4 Stochastic Processes with an (V)ARMA Background Process

ARMA processes are very flexible in modeling autocorrelation, e.g. an $AR(p)$ can model $p$ lags of autocorrelation exactly. However, the processes as defined in Eq. 5 result in a normal marginal distribution, which makes them not really suitable for simulation input modeling in most cases where the distribution is clearly non-normal.

A promising approach to overcome this limitation is to use ARMA processes as a background process for only modeling the autocorrelation and combine them with an arbitrary marginal distributions.

This idea has been introduced by ARTA processes [7] that combine an $AR(p)$ base process as defined in Eq. 5 with a marginal distribution $F$ using the inversion method by setting $Y_t = F^{-1}[\Phi(Z_t)], (t = 1, 2, \ldots)$ where $\Phi$ is the standard normal cumulative distribution function. The $AR(p)$ process is constructed such that the distribution of the $\{Z_t\}$ is $N(0,1)$,

resulting in $\Phi(Z_t)$ to have uniform distribution on $(0, 1)$. The inverse transformation finally yields a time series $Y_t$ with the desired marginal distribution. Since the $Z_t$ are autocorrelated, the $Y_t$ are as well, and it is possible to establish a relation between the two autocorrelations.

In [2] ARTA processes have been generalized to Vector ARTA (VARTA) processes that can model multivariate time series. In this case the base process is a $VAR(p)$ process as defined in Eq. 6 and the VARTA process $\boldsymbol{Y}_t = (Y_t^{(1)}, Y_t^{(2)}, \ldots, Y_t^{(k)})$ is obtained by $Y_t^{(i)} = F_{(i)}^{-1}(\Phi(Z_t^{(i)})), i = 1, \ldots, k$. [2] assumed that the $F_{(i)}$ are marginal distributions from the Johnson system of distributions though other distributions are possible as well of course.

(V)ARTA processes rely on the inversion of the distribution function and therefore are applicable for all marginal distributions $F$ for which a closed-form expression for the inverse cdf exists or for which $F^{-1}$ can be computed numerically in an efficient way. In practice some interesting and useful distributions cannot be used as part of ARTA processes because of this. PHDs from Sect. 2.2 have proven to be suitable distributions when it comes to modeling complicated empirical distributions from real-world observations that for example have been recorded in computer or communication networks [6]. However, as we can see from Eq. 2 the cumulative distribution function of CPHDs contains a matrix exponential which makes it in general inefficient and difficult to numerically compute the inverse of the cdf. [17] presented a different approach to combine acyclic CPHDs and ARMA processes, denoted as CAPP (Correlated Acyclic Phase-Type Process), that will be used as basis for the work in this paper. The key idea is to interpret the CPHD as a set of paths from an initial state (i.e. states $i$ with $\boldsymbol{\pi}(i) > 0$) to the absorbing state and use a background ARMA process to choose the next path when restarting after absorption.

In the following we will extend CAPPs in two directions. First, we will integrate acyclic DPHDs into the process, which will be helpful to model measurements with discrete values. Second, we will generalize the process description to the multivariate case to allow for correlation between the different variates.

# 3 Vector Correlated Acyclic Phase-Type Processes

In the following we present our approach to combine $k$ acyclic PHDs (discrete and continuous) with a VARMA background process, denoted as Vector Correlated Acyclic Phase-Type Process $VCAPP_k(n_1, \ldots, n_k, p, q)$, where $n_1, \ldots, n_k$ define the number of transient states of the PHDs and $p, q$ give the order of the VARMA base process.

We first describe how the PHDs are represented as paths from an initial to the absorbing state (called elementary series), then show how the PHDs and the VARMA process are combined, how we can compute basic properties of the process and finally present algorithms for parameter estimation and random number generation for the process class.

## 3.1 Splitting Acyclic PHDs into Paths

As already mentioned we express a PHDs in terms of its elementary series, where each series describes one path from an initial state to the absorbing state. For CPHDs this concept has been introduced in [10] and it was later extended to DPHDs in [3].

For an acyclic CPHD $(\boldsymbol{\pi}, \boldsymbol{D}_0)$ the $i$-th series is described by a vector $\boldsymbol{\Lambda_i}$ that contains the transition rates of the states of the

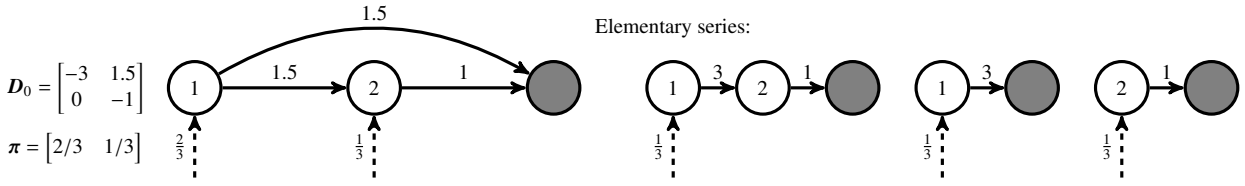
series and an associated probability $v_i$ that is computed from the the transition rates along the path and the initial probability of the first state of the path, i.e. $v_i$ is the probability that this path is chosen. More formally, let $i_1, i_2, \ldots, i_k$ be the states of an elementary series. Then the probability of this series is given by

$$v_i = \boldsymbol{\pi}(i_1) \frac{\boldsymbol{D}_0(i_1, i_2)}{-\boldsymbol{D}_0(i_1, i_1)} \frac{\boldsymbol{D}_0(i_2, i_3)}{-\boldsymbol{D}_0(i_2, i_2)} \cdots \frac{\boldsymbol{d}_1(i_k)}{-\boldsymbol{D}_0(i_k, i_k)}$$

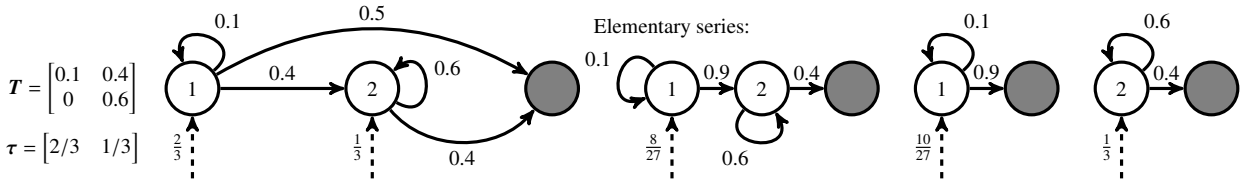
where $\boldsymbol{d}_1(i_k)$ denotes the transition rate from state $i_k$ to the absorbing state. The vector of transition rates is given by

$$\boldsymbol{\Lambda_i} = (-\boldsymbol{D}_0(i_1, i_1), -\boldsymbol{D}_0(i_2, i_2), \ldots, -\boldsymbol{D}_0(i_k, i_k)).$$

An example for an acyclic CPHD and its elementary series is given in Fig. 1a. In the discrete case the $i$-th series of a PHD




(a) CPHD




(b) DPHD

Figure 1: Representing acyclic PHDs as Elementary Series

$(\boldsymbol{\tau}, \boldsymbol{T})$ is represented by a vector $\boldsymbol{P}_i$ with transition probabilities of the states of the series and the associated probability $\phi_i$. Now we have for a series of states $i_1, i_2, \ldots, i_k$ that

$$\boldsymbol{P}_i = (\boldsymbol{T}(i_1, i_1), \ldots, \boldsymbol{T}(i_k, i_k)) \text{ and } \phi_i = \frac{\boldsymbol{\tau}(i_1)\boldsymbol{T}(i_1, i_2)}{1 - \boldsymbol{T}(i_1, i_1)} \frac{\boldsymbol{T}(i_2, i_3)}{1 - \boldsymbol{T}(i_2, i_2)} \cdots \frac{\boldsymbol{t}(i_k)}{1 - \boldsymbol{T}(i_k, i_k)}$$

where $\boldsymbol{t}(i_k)$ denotes the transition probability from state $i_k$ to the absorbing state. An example for an acyclic DPHD and its elementary series is given in Fig. 1b.

Note, that an elementary series describes a generalized Erlang (or hypoexponential) distribution in the continuous case and a sequence of geometric distributions in the discrete case.

## 3.2 Combining PHDs and VARMA Process

Now assume, that we have $k$ PHDs each expressed in terms of their elementary series, i.e. for PHD $i$ we have the series $j = 1, \ldots, m_i$ each consisting of an initial probability $v_{i_j}$ and the transition rates $\boldsymbol{\Lambda_{i_j}}$ in the continuous case or a probability $\phi_{i_j}$ and vector $\boldsymbol{P}_{i_j}$ in the discrete case. To simplify notations we denote the probability of an elementary series $j$ of PHD $i$

with $\psi_{i_j}$ if it is not important whether it is a series from a continuous or discrete distribution. We additionally introduce a stationary $VARMA(p,q)$ base process $\boldsymbol{Z}_t$ as defined in Eq. 6 that is used to choose between these series, thereby introducing correlation into the sequence of phase-type distributed random variables. More formally, we use the following definitions to describe VCAPPs: For the elementary series of PHD $i$ we define

$$\begin{aligned} \underline{b}_{i_1} &= 0 \\ \bar{b}_{i_j} &= \underline{b}_{i_j} + \psi_{i_j} \qquad j = 1, \ldots, m_i \qquad \text{and} \qquad \delta(U,i,j) = \begin{cases} 1, & U \in [\underline{b}_{i_j}, \bar{b}_{i_j}) \\ 0, & \text{otherwise} \end{cases} \\ \underline{b}_{i_j} &= \bar{b}_{i_{j-1}} \qquad j = 2, \ldots, m_i \end{aligned} \tag{7}$$

for some random variable $U$ with uniform distribution. Let $\{X_t^{(i,j)}\}$ be sequences of iid random variables that either have a generalized Erlang distribution described by the rates in $\boldsymbol{\Lambda_{i_j}}$ if $i$ is a CPHD or describe a sequence of geometric distributions as defined by the probabilities in $\boldsymbol{P_{i_j}}$ if $i$ is discrete.

Using the above definitions a $VCAPP_k(n_1, \ldots, n_k, p, q)$ describes a time-series $\boldsymbol{Y}_t = (Y_t^{(1)}, Y_t^{(2)}, \ldots, Y_t^{(k)})'$ where each $Y_t^{(i)}$ is defined as

$$Y_t^{(i)} = \sum_{j=1}^{m_i} \delta(\Phi(Z_t^{(i)}), i, j) X_t^{(i,j)}. \tag{8}$$

The $Z_t^{(i)}$ are generated by a $VARMA(p,q)$ process as defined in Eq. 6 and we require them to have standard normal marginal distribution. $\Phi(\cdot)$ is the standard normal cumulative distribution function which implies that $\Phi(Z_t^{(i)})$ have uniform distribution on $(0,1)$ (cf. [11]). Then Eq. 8 uses the elementary series to describe a sequence of correlated random variables with the same acyclic PHD that the elementary series have been computed from (see also [17]).

In the following we relate the VARMA base process and the VCAPP autocorrelation and describe a procedure to construct a base process that results in the desired VCAPP correlation and that respects the requirements pointed out above (i.e. a stable process where the the $Z_t^{(i)}$ have standard normal marginal distribution).

### 3.3 Computation of the VCAPP Autocorrelation

Assume that a $VCAPP_k(n_1, \ldots, n_k, p, q)$ is given. The autocorrelation and autocovariance of a $VARMA(p,q)$ process can be expressed in terms of the autocorrelation and autocovariance matrices. Note, that we required the $Z_t^{(i)}$ to have standard normal marginal distribution and thus, autocorrelation and autocovariance are identical for the base process. In the following we will sketch how to compute the autocorrelations of the $VARMA(p,q)$ base process and then relate the base process correlation with the $VCAPP_k(n_1, \ldots, n_k, p, q)$ correlation.

The theoretical background and the methods for the computation of the $VARMA(p,q)$ correlation are given in [21]. We will only summarize the basic steps. The key idea of the computation is the transformation of the $VARMA(p,q)$ into an equivalent $VAR(1)$ representation, from which the correlation can be computed easily. Let $\{\boldsymbol{Z}_t\}$ be a stable $k$-variate $VARMA(p,q)$ process as defined in Eq. 6. We denote the equivalent $VAR(1)$ process $\tilde{\boldsymbol{Z}}_t = \tilde{\boldsymbol{\alpha}}\tilde{\boldsymbol{Z}}_{t-1} + \tilde{\boldsymbol{\epsilon}}_{\boldsymbol{t}}$. The construction of the matrices is summarized in [18, Sect. 2] (see also [21] for the details). For the $VAR(1)$ representation $\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{Z}}}(0)$, i.e. the

covariance matrix at lag 0 can be easily obtained from

$$vec(\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{Z}}}(0)) = (\boldsymbol{I} - \tilde{\boldsymbol{\alpha}} \otimes \tilde{\boldsymbol{\alpha}})^{-1} \, vec(\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}) \tag{9}$$

where the $vec()$ operator transforms a matrix into a vector by stacking the columns. $\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{Z}}}(0)$ now contains the desired covariance matrices $\boldsymbol{\Sigma}_Z(h)$ for $h = 0, \ldots, p-1$ of the original $VARMA(p,q)$ process as submatrices (i.e. $\boldsymbol{\Sigma}_Z(h)$ is a block in the upper left, followed by $\boldsymbol{\Sigma}_Z(1)$ and so on) [21]. If $\boldsymbol{\Sigma}_Z(h)$ for $h = 0, \ldots, p-1$ is known higher lags can be computed recursively from [21]

$$\boldsymbol{\Sigma}_Z(h) = \boldsymbol{\alpha_1}\boldsymbol{\Sigma}_Z(h-1) + \cdots + \boldsymbol{\alpha_p}\boldsymbol{\Sigma}_Z(h-p). \tag{10}$$

The computation of the autocovariances requires $p > q$. Therefore, if $p \leq q$ we have to add additional matrices $\boldsymbol{\alpha}_i = 0$ before the transformation into the $VAR(1)$ process. The autocorrelations can be obtained from the autocovariances by $\boldsymbol{\rho}_Z(h) = \boldsymbol{D}^{-1}\boldsymbol{\Sigma}_Z(h)\boldsymbol{D}^{-1}$ where $\boldsymbol{D}$ is a diagonal matrix that has the square roots of the diagonal elements of $\boldsymbol{\Sigma}_Z(0)$ as elements. Recall, that this last step is not necessary in our case, because we required the $\boldsymbol{Z}_t$ to have standard normal distribution. Thus, $\boldsymbol{D}^{-1} = I$ and $\boldsymbol{\Sigma}_Z(h) = \boldsymbol{\rho}_Z(h)$.

Once the $VARMA(p,q)$ autocorrelation matrices are known, we can use them to compute the autocorrelation matrices of the VCAPP. Let $\boldsymbol{\rho}_Z(h)$ be a matrix with autocorrelation coefficients at lag $h$ from a $VARMA(p,q)$ process computed as described above. Then, we want to determine matrix $\boldsymbol{\rho}_Y(h)$ that contains the corresponding correlation coefficients of the VCAPP. Furthermore, let $\rho_Z(i,j,h)$ and $\rho_Y(i,j,h)$ be the elements at position $(i,j)$ of $\boldsymbol{\rho}_Z(h)$ and $\boldsymbol{\rho}_Y(h)$, respectively. Then,

$$\rho_Y(i,j,h) = Corr[Y_t^{(i)}, Y_{t+h}^{(j)}] = \frac{E[Y_t^{(i)}Y_{t+h}^{(j)}] - E[Y^{(i)}]E[Y^{(j)}]}{\sqrt{Var[Y^{(i)}]Var[Y^{(j)}]}} \tag{11}$$

Since $E[Y^{(i)}]$, $E[Y^{(j)}]$, $Var[Y^{(i)}]$ and $Var[Y^{(j)}]$ are fixed by the acyclic PHDs $i$ and $j$, i.e. they can be computed using Eq. 2 or Eq. 4, the only remaining part when computing the VCAPP correlation is the joint moment $E[Y_t^{(i)}Y_{t+h}^{(j)}]$. After some substitutions and rearranging we get (cf. [17])

$$\begin{aligned}
E[Y_t^{(i)}Y_{t+h}^{(j)}] &= E\left[\left(\sum_{k=1}^{m_i}\delta(\Phi(Z_t^{(i)}),i,k)X_t^{(i,k)}\right)\left(\sum_{l=1}^{m_j}\delta(\Phi(Z_{t+h}^{(j)}),j,l)X_{t+h}^{(j,l)}\right)\right] \\
&= \sum_{k,l} E\left[\delta(\Phi(Z_t^{(i)}),i,k)\delta(\Phi(Z_{t+h}^{(j)}),j,l)\right] E\left[X_t^{(i,k)}\right] E\left[X_{t+h}^{(j,l)}\right] \qquad (12) \\
&= \sum_{k,l}\left(\mu^{(i,k)}\mu^{(j,l)}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\delta(\Phi(Z_t^{(i)}),i,k)\delta(\Phi(Z_{t+h}^{(j)}),j,l)\right. \\
&\qquad \left.\varphi_{\rho_Z(i,j,h)}(z_t^{(i)}, z_{t+h}^{(j)})dz_t^{(i)}dz_{t+h}^{(j)}\right) \\
&= \sum_{k,l}\left(\mu^{(i,k)}\mu^{(j,l)}\int_{\Phi^{-1}(\underline{b}_{i_k})}^{\Phi^{-1}(\bar{b}_{i_k})}\int_{\Phi^{-1}(\underline{b}_{j_l})}^{\Phi^{-1}(\bar{b}_{j_l})}\varphi_{\rho_Z(i,j,h)}(z_t^{(i)}, z_{t+h}^{(j)})dz_t^{(i)}dz_{t+h}^{(j)}\right) \qquad (13)
\end{aligned}$$

where $\mu^{(i,k)}$ is the first moment of the $k$-th elementary series of the $i$-th PHD. Since the series of a PHD are either a generalized Erlang distribution or a sequence of geometric distributions, the series are, of course, simple PHDs themselves and the moments can be easily computed according to Eqs. 2 and 4. $\varphi_{\rho_Z(i,j,h)}(z_t^{(i)}, z_{t+h}^{(j)})$ is the bivariate standard normal density function with correlation $\rho_Z(i,j,h)$. Note, that Eq. 12 holds because in our process description the base process

is used to determine which series of the PHDs are taken but the duration in that series is independent of the base process. Eq. 13 holds because $\delta(u,i,k)$ is 1 for $u \in [\underline{b}_{i_k}, \bar{b}_{i_k})$ and 0 otherwise (cf. Eq. 7) and we can exploit this information to determine the integration bounds. For the computation of the bivariate normal integral fast numerical procedures exist [12]. Observe from Eq. 13 that for a given PHD the VCAPP correlation only depends on the VARMA correlation which appears in $\varphi_{\rho_Z(i,j,h)}(z_t^{(i)}, z_{t+h}^{(j)})$. Therefore, we may express $\rho_Y(i,j,h)$ as a function of $\rho_Z(i,j,h)$, i.e. $\rho_Y(i,j,h) = \omega(\rho_Z(i,j,h))$. In the following we assume that the elementary series of a PHD $i$ are sorted according to their mean values $\mu^{(i,k)}$. Then $\omega(\cdot)$ has the following properties that are useful for the construction of the base process in Sect. 3.4 and that we state without proof: $\omega(\cdot)$ is a continuous and non-decreasing function. This immediately implies that the maximal and minimal possible autocorrelations $\hat{\rho}_{max}$ and $\hat{\rho}_{min}$ for a VCAPP are given by $\hat{\rho}_{max} = \omega(1)$ and $\hat{\rho}_{min} = \omega(-1)$, respectively. Furthermore we have that $\omega(0) = 0$ and $\rho_Z(i,j,h) \leq 0\ (\geq 0) \Rightarrow \omega(\rho_Z(i,j,h)) \leq 0\ (\geq 0)$.

## 3.4 An Algorithm for Fitting VCAPPs

Using the considerations from Sect. 3.3 we can sketch the algorithm from Fig. 2 for fitting VCAPPs from observations from a real system. As inputs the algorithm takes a $k$-variate trace $\mathcal{T}^{(k)}$ and the order $p, q$ of the base process. In the first step the

```
Inputs:
  k-variate trace T^(k); order of base process: p,q

1) fit k PHDs to the k sequences of T^(k)
2) compute correlation matrices ρ̂_Y(i), i = 1,...,H from T^(k) using Eq. 1
3) determine and sort elementary series of APH(n_i), i = 1,...,k
4) determine VARMA autocorrelations ρ̂_Z(i), i = 1,...,H such that VCAPP has
     correlation ρ̂_Y(i), i = 1,...,H using Eq. 13 and a search algorithm
5a) if (q==0) and (p=H): Compute VAR(p) base process according to Eq. 14
5b) otherwise: minimize Eq. 16 to find a VARMA(p,q) model for ρ̂_Z(i), i = 1,...,H
6) return VCAPP_k(n_1,...,n_k,p,q) with base VARMA(p,q) process and k PHDs
```

Figure 2: Basic Steps for Parameter Estimation of VCAPPs

algorithm fits a PHD to each of the sequences of $\mathcal{T}^{(k)}$ using one of the available approaches mentioned in Sect. 2.2. Next the correlation matrices $\hat{\boldsymbol{\rho}}_Y(i), i = 1, \ldots, H$ are estimated from the multivariate trace which describe the desired correlation the VCAPP should have. After that the algorithm determines the elementary series of the acyclic PHDs as described in Sect. 3.1 and sorts them according to their mean values. In the next step the VARMA correlation $\hat{\boldsymbol{\rho}}_Z(i), i = 1, \ldots, H$ that yields a VCAPP correlation $\hat{\boldsymbol{\rho}}_Y(i), i = 1, \ldots, H$ has to be determined. Observe from Eq. 13 that we can compute a $\hat{\rho}_Y(i,j,h)$ for a given $\hat{\rho}_Z(i,j,h)$ but not the other way round, which is necessary for constructing the VARMA base process. Consequently, we have to determine the base process autocorrelation numerically using Eq. 13 which can be done using a simple line search algorithm [24], since $\omega(\cdot)$ is a continuous and non-decreasing function. It should be noted that also for the original CAPP approach and the (V)ARTA approaches a numerical procedure has to be applied for computing the base process autocorrelation [7, 2, 17].

Once the base process autocorrelations $\hat{\boldsymbol{\rho}}_Z(h)$ have been determined we have to construct a $VARMA(p,q)$ base process that exhibits this structure. Depending on whether the base process is a $VARMA(p,q)$ process or a $VAR(p)$ process without moving average coefficients this can be done using a general purpose optimization algorithm or by solving Yule-Walker equations. Fitting a $VAR(p)$ is very fast but might result in a large base process because it requires $p = H$, where $H$ equals the number of autocorrelation lags that are considered, i.e. we need a matrix with autoregressive coefficients for each correlation matrix. Fitting a $VARMA(p,q)$ process can result in a smaller model but the parameter estimation is more elaborate.

A $VAR(p)$ model can be obtained by solving the the Yule-Walker equations [21], i.e. $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p)$ is obtained from

$$\boldsymbol{\alpha} = \boldsymbol{\Sigma}{\boldsymbol{\Sigma}_{\boldsymbol{Z}}}^{-1} \tag{14}$$

where $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_Z(1), \boldsymbol{\Sigma}_Z(2), \ldots, \boldsymbol{\Sigma}_Z(p))$ and

$$\boldsymbol{\Sigma}_{\boldsymbol{Z}} = \begin{bmatrix} \boldsymbol{\Sigma}_Z(0) & \boldsymbol{\Sigma}_Z(1) & \cdots & \boldsymbol{\Sigma}_Z(p-2) & \boldsymbol{\Sigma}_Z(p-1) \\ \boldsymbol{\Sigma}'_Z(1) & \boldsymbol{\Sigma}_Z(0) & \cdots & \boldsymbol{\Sigma}_Z(p-3) & \boldsymbol{\Sigma}_Z(p-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{\Sigma}'_Z(p-1) & \boldsymbol{\Sigma}'_Z(p-2) & \cdots & \boldsymbol{\Sigma}'_Z(1) & \boldsymbol{\Sigma}_Z(0) \end{bmatrix}. \tag{15}$$

Once the $\boldsymbol{\alpha}_i$ are known we compute $\boldsymbol{\Sigma}_\epsilon = \boldsymbol{\Sigma}_Z(0) - \boldsymbol{\alpha}_1\boldsymbol{\Sigma}'_Z(1) - \cdots - \boldsymbol{\alpha}_p\boldsymbol{\Sigma}'_Z(p)$. If we set $\boldsymbol{\Sigma}_Z(i) = \hat{\boldsymbol{\rho}}_Z(i)$ in the above equations, i.e. we assume that the autocovariance equals the autocorrelation, we get a process with the desired property to have standard normal marginal distribution.

When fitting a $VARMA(p,q)$ process to $\hat{\boldsymbol{\rho}}_Z(h)$ we use the Nelder-Mead algorithm from [22] to minimize

$$\underset{\boldsymbol{\alpha}_i, i=1,\ldots,p, \boldsymbol{\beta}_j, j=1,\ldots,q}{\arg\min} \sum_{l=1}^{k}\sum_{m=1}^{k}\sum_{h=1}^{H}\left(\frac{\rho_Z(l,m,h)^*}{\hat{\rho}_Z(l,m,h)} - 1\right)^2 \tag{16}$$

where $k$ is the number of variates, $H$ is the number of autocorrelation lags to consider for fitting, $\hat{\rho}_Z(l,m,h)$ is the desired autocorrelation of the base process and $\rho_Z(l,m,h)^*$ is the autocorrelation of the VARMA process constructed during the fitting process. Since we are only interested in stationary base processes, non-stationary solutions have to be penalized in the fitting step. According to [21] stability of a VARMA process implies stationarity and a VARMA process is stable if all roots of the reverse characteristic polynomial $|\boldsymbol{I}_{k\times k} - \boldsymbol{\alpha}_1 z - \boldsymbol{\alpha}_2 z^2 - \cdots - \boldsymbol{\alpha}_p z^p| = 0$ lie outside the unit circle. Hence, for VARMA processes that are constructed during the fitting process and that do not fulfill this requirement we add a penalty term to Eq. 16.

Observe, that the covariance matrix of the innovations $\boldsymbol{\Sigma}_\epsilon$ is not part of the minimization in Eq. 16. Instead we set the values of $\boldsymbol{\Sigma}_\epsilon$ separately, such that the VARMA process has standard normal marginal distributions. For given AR and MA coefficient matrices we use Eq. 9 to construct a system of linear equations to determine the entries of $\boldsymbol{\Sigma}_\epsilon$. This requires $\boldsymbol{\Sigma}_\epsilon$ to be a diagonal matrix. Restricting the structure of the covariance matrix is not uncommon to cope with the complexity of VARMA processes [8]. Details on the construction of $\boldsymbol{\Sigma}_\epsilon$ can be found in [18, Sect. 3].

The minimization can be performed for different values of $p$ and $q$ and the best model according to Eq. 16 is selected. Of course, one has to find a compromise between a moderate size of the parameters $p$ and $q$ to keep the model size small and

an adequate fitting of the lag $h$ ($h = 1, \ldots, H$) correlations. Finally, the PHDs and the $VARMA(p, q)$ are combined into a $VCAPP_k(n_1, \ldots, n_k, p, q)$.

### 3.5 Generating Random Numbers from VCAPPs

The generation of multivariate samples from a $VCAPP_k(n_1, \ldots, n_k, p, q)$ can be efficiently performed in two steps. In the first step a multivariate sample from the $VARMA(p, q)$ base process is generated using standard theory [21]. In the second step this sample is used to generate the VCAPP random vector.

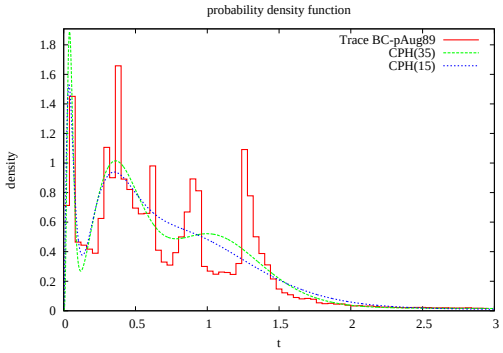
From the $VARMA(p, q)$ description the coefficient matrices $\boldsymbol{\alpha}_i, i = 1, \ldots, p$ and $\boldsymbol{\beta}_j, j = 1, \ldots, q$ and the covariance matrix $\boldsymbol{\Sigma}_\epsilon$ for the innovations are known. Furthermore, we assume that the previous observations $\boldsymbol{z}_{t-1}, \ldots, \boldsymbol{z}_{t-p}$ and the previous innovations $\boldsymbol{\epsilon_{t-1}}, \ldots, \boldsymbol{\epsilon_{t-q}}$ are known, either from previous simulation steps of the $VARMA(p, q)$ process or from an initialization step that is described in [21] and summarized in [18, Sect. 4]. Then, the next random sample can be determined recursively using Eq. 6. First, we compute the next vector with innovations $\boldsymbol{\epsilon_t}$. This is done by determining matrix $\boldsymbol{S}$ with $\boldsymbol{SS'} = \boldsymbol{\Sigma}_\epsilon$ applying a Cholesky decomposition [13]. Now, $\boldsymbol{\epsilon_t} = \boldsymbol{S}\boldsymbol{v}_t$ where $\boldsymbol{v}'_t = (v_1, \ldots, v_k)$ contains $k$ random numbers drawn from a standard normal distribution. Since now all values from Eq. 5 are known we can compute $\boldsymbol{z}_t$ according to that definition.

In the second step of the random number generation procedure the $\boldsymbol{z}_t$ are used to determine the VCAPP random vector $\boldsymbol{y}_t$ using Eq. 8. For each $\boldsymbol{y}_t^{(i)}$ we compute $\Phi(\boldsymbol{z}_t^{(i)})$, determine the interval $j$ for which $\Phi(\boldsymbol{z}_t^{(i)}) \in [\underline{b}_{i_j}, \bar{b}_{i_j}]$. If PHD $i$ is continuous we draw from an exponential distribution for each rate from $\boldsymbol{\Lambda_i}$. The sum of these exponentially distributed random numbers is returned as $\boldsymbol{y}_t^{(i)}$. If PHD $i$ is discrete we draw from the geometric distributions that are defined by $\boldsymbol{P}_i$ and return the sum of the number of trials as $\boldsymbol{y}_t^{(i)}$.

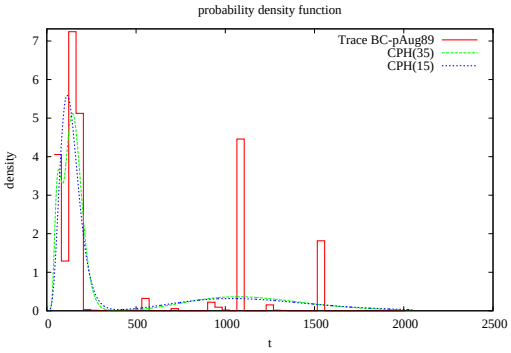
## 4 Experimental Results

We conducted two series of experiments to show the performance of VCAPPs when fitting real traffic data. For the first experiments we used the well known trace BC-pAug89 [20] that is often used as a benchmark trace and that contains the interarrival times and packet sizes of 1 million packets in a local area network. The second more recent trace was recorded at Dartmouth College [15] and additionally includes user mobility in a wireless network.

As mentioned above BC-pAug89 is a 2-variate trace consisting of one sequence for the interarrival times of packets and another sequence with the corresponding packet sizes. We fitted CPHDs with different number of states to both of the sequences using the approach from [28]. The results for CPHDs with 15 and 35 states are shown in Figs. 3a and 3b. As we can see the CPHDs provided a good approximation for the interarrival times, but only were able to capture the first peak of the packet sizes. According to the algorithm from Fig. 2 we expanded the CPHDs into two VCAPPs. The larger VCAPPs consists of the two CPHDs with 35 states and a $VAR(20)$ base process, the smaller VCAPP of the two CPHDs with 15 states and a $VAR(5)$ base process. Since we used VAR processes the VCAPPs provided an exact matching of 5 and 20




(a) pdf of interarrival times




(b) pdf of packet sizes

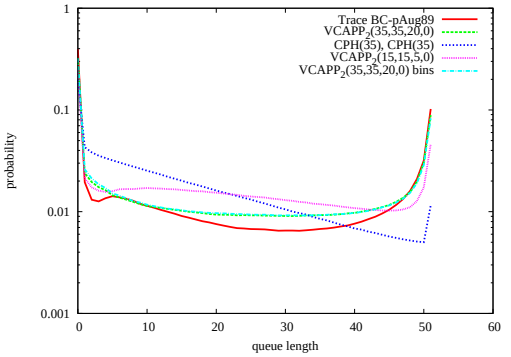
$$\hat{\rho}_Y(0) = \begin{bmatrix} 1.0 & -0.1 \\ -0.1 & 1.0 \end{bmatrix}$$

$$\hat{\rho}_Y(1) = \begin{bmatrix} 0.2 & -0.1 \\ -0.15 & 0.45 \end{bmatrix}$$

$$\hat{\rho}_Y(2) = \begin{bmatrix} 0.19 & -0.06 \\ -0.11 & 0.41 \end{bmatrix}$$

$$\hat{\rho}_Y(3) = \begin{bmatrix} 0.14 & -0.02 \\ -0.09 & 0.36 \end{bmatrix}$$

(c) correlation matrices



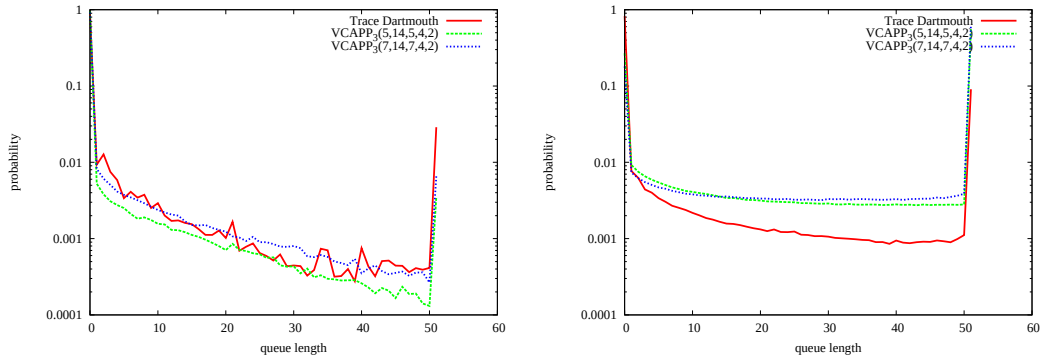

(d) queue length distribution

Figure 3: Results for Trace BC-pAug89

lags of correlation, respectively. The correlation matrices for the first lags are shown in Fig. 3c. As a last measure for the quality of the fitting we determined the queue length distribution of a single server queue with a buffer capacity of 50. The VCAPPs are used to obtain the interarrival times of the packets and the exponentially distributed service time depends on the packet sizes also obtained from the VCAPPs. To obtain reference values we also simulated the model with the original trace values. Fig. 3d shows the queue length distribution obtained from 10 replications and approx. 1 million generated packets per replication. As we can see the large $VCAPP_2(35, 35, 20, 0)$ provides a very good approximation of the trace results, while for the smaller $VCAPP_2(15, 15, 5, 0)$ the results are not as good. For comparison we also generated results from the two CPHDs of the larger VCAPP without using the base process, i.e. the generated values are uncorrelated. The curve shows that neglecting the correlation significantly impairs the quality of the approximation. Finally, we used a modified version of the $VCAPP_2(35, 35, 20, 0)$ where we sorted the packets into bins of size 50 bytes resulting in a much smaller number of discrete values and used a DPHD for fitting. For the simulation each bin was represented by its midpoint. As shown in Fig. 3d the curves for the original $VCAPP_2(35, 35, 20, 0)$ and the variant with bins almost completely overlap.

At Dartmouth College large traces of data have been recorded from their wireless network over the past years. The traces contain movement data and tcpdump data with generated traffic. The movement data consists of a trace for every user seen in

the network with dwell times and assignments to different access points. A summary of the data and collection methods can be found in [15]. Out of this data we generated a 3-variate trace consisting of the dwell times and locations from the movement data and the generated traffic from the tcpdump data. Unfortunately the movement data was recorded for a larger period of time than the tcpdump data. Moreover, tcpdump data does not exist for all access points occurring in the movement data. To obtain suitable traces for fitting we preprocessed the data in the following way: We parsed every movement trace and for each entry in the trace we checked the tcpdump data if packets have been recorded for that user in the time period specified by the movement data and summed up the amount of traffic from all packets in that period. Since traffic information is not available for all entries of the movement data we extracted consecutive sequences of at least 100 entries from the movement data for that the traffic amount could be computed. Hence, we obtained traces of the form $\mathcal{T}_i = ((t_{i,1}, l_{i,1}, p_{i,1}), (t_{i,2}, l_{i,2}, p_{i,2}), \cdots)$ describing the behavior of $i = 1, \cdots, 501$ users in the network where $t_{i,j}, l_{i,j}$ and $p_{i,j}$ are the $j$-th dwell time, location and amount of traffic of user $i$, respectively. Thus, we generated traces with realistic movement patterns, although the traces of course do not fully reflect the original situation at Dartmouth College because some original entries had to be ignored and some movements of a single user have been split into several traces. As locations we used 14 different building on the campus (i.e. all access points that lie in the same building are aggregated into a single location). The locations were fitted with a DPHD with 14 states such that we can ensure that only values from 1 to 14 can be drawn from that distribution. The CPHDs for dwell times and traffic are again fitted with the approach from [28]. For distribution fitting we can just combine all traces $\mathcal{T}_i$ under the assumptions that the users behave identically. For the computation of the correlation matrices this is not possible. Therefore, we computed correlation matrices $\hat{\boldsymbol{\rho}}_i(h)$ for each trace $\mathcal{T}_i$ and then weighted them according to the number of entries, i.e. $\hat{\boldsymbol{\rho}}(h) = \sum_i r_i/r \cdot \hat{\boldsymbol{\rho}}_i(h)$, where $r_i$ is the number of entries in $\mathcal{T}_i$ and $r$ is the total number of entries in all traces and fitted VCAPPs according to $\hat{\boldsymbol{\rho}}(h)$. To assess the fitting quality we set up a model with 25 users each driven by a




(a) queue length dist. for router 4

(b) queue length dist. for router 7

Figure 4: Results for Trace Dartmouth

VCAPP. For each of the locations we use a router with buffer size 50. The random samples from the VCAPP then determine what amount of traffic a user generates at a location/router and how long he stays there before moving to the next location. As reference values we again simulated the model using the trace files. Since we have more traces than users in the model each user randomly selects a trace for the simulation and we run the model with 100 replications. The queue length distributions

for the traces and two different VCAPPs of two of the routers are shown in Fig. 4. As we can see we obtained a very good approximation for some routers as in Fig. 4a, while for other routers the approximation was not that good but still acceptable (Fig. 4b).

# 5 Conclusions

We presented an approach for parameter estimation of multivariate traces that uses PHDs for fitting the marginal distributions of the trace and a VARMA process for adding correlation. The approach divides the complex problem of fitting multivariate traces into several steps that can be performed independent of each other, i.e. the PHDs determine the correlation matrices of the VARMA process, but have no further influence when constructing the process. Our experiments with different traces suggest that the process is suitable for approximating real traffic data. For the wireless data it is probably possible to further improve the results by dividing the users into classes according to their behavior and fit a VCAPP for each of the classes, but this is subject to further research.

# References


[1] B. Biller and C. Gunes. Introduction to simulation input modeling. In *Proc. of the Winter Simulation Conference*, 2010.

[2] B. Biller and B. L. Nelson. Modeling and Generating Multivariate Time-Series Input Processes using a Vector Autoregressive Technique. *ACM Transactions on Modeling and Computer Simulation*, 13(3):211–237, 2003.

[3] A. Bobbio, A. Horváth, M. Scarpa, and M. Telek. Acyclic discrete phase type distributions: properties and a parameter estimation algorithm. *Perf. Eval.*, 2003.

[4] G. Box and G. Jenkins. *Time series analysis: Forecasting and control*. Holden-Day, San Francisco, 1970.

[5] P. Buchholz and J. Kriege. A Heuristic Approach for Fitting MAPs to Moments and Joint Moments. In *Proc. of QEST*, pages 53–62, 2009.

[6] P. Buchholz, J. Kriege, and I. Felko. *Input Modeling with Phase-Type Distributions and Markov Models - Theory and Applications*. Springer, 2014.

[7] M. C. Cario and B. L. Nelson. Autoregressive to anything: Time-series input processes for simulation. *Operations Research Letters*, 19(2):51–58, 1996.

[8] J. C. Chan and E. Eisenstat. Gibbs Samplers for VARMA and Its Extensions. Technical report, Australian National University, School of Economics, 2013.

[9] I. Civelek, B. Biller, and A. Scheller-Wolf. The impact of dependence on queueing systems. Research Showcase 8-2009, Carnegie Mellon University, 2009.

[10] A. Cumani. On the Canonical Representation of Homogeneous Markov Processes Modeling Failure-Time Distributions. *Micorelectronics and Reliability*, 1982.

[11] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, New York, 1986.

[12] Z. Drezner and G. O. Wesolowsky. On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, 35:101 – 107, 1990.

[13] G. S. Fishman. *Concepts and Methods in Discrete Event Digital Simulation*. John Wiley, New York, 1973.

[14] S. Fu and C. Xu. Quantifying temporal and spatial correlation of failure events for proactive management. In *Proc. of SRDS*, 2007.

[15] T. Henderson, D. Kotz, and I. Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In *Proc. of MobiCom*, 2004.

[16] R. E. A. Khayari, R. Sadre, and B. Haverkort. Fitting World-Wide Web Request Traces with the EM-Algorithm. *Performance Evaluation*, 52:175–191, 2003.

[17] J. Kriege and P. Buchholz. Correlated phase-type distributed random numbers as input models for simulations. *Performance Evaluation*, 68(11):1247–1260, 2011.

[18] J. Kriege and P. Buchholz. Online Companion to the Paper 'Traffic Modeling with Phase-Type Distributions and VARMA Processes', 2016.

[19] A. M. Law. *Simulation Modleing and Analysis*. Mc Graw Hill, 2013.

[20] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the Self-Similar Nature of Ethernet Traffic. *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.

[21] H. Lütkepohl. *Introduction to Multiple Time Series Analysis*. Springer, 1993.

[22] J. Nash. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation*. Adam Hilger, 2. edition, 1990.

[23] M. F. Neuts. A versatile Markovian point process. *Journ. of Appl. Prob.*, 1979.

[24] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1993.

[25] P. Reinecke and G. Horváth. Phase-type distributions for realistic modelling in discrete-event simulation. In *Proc. of SimuTools*, 2012.

[26] R. D. Smith. The dynamics of internet traffic: Self-similarity, self-organization, and complex phenomena. *Advances in Complex Systems*, 14(6):905–949, 2011.

[27] P. Stanfield, J. Wilson, and R. King. Flexible modelling for correlated operation times with application in product-reuse facilities. *Int. Journ. of Prod. res.*, 2004.

[28] A. Thümmler, P. Buchholz, and M. Telek. A Novel Approach for Phase-Type Fitting with the EM Algorithm. *IEEE Trans. on Dep. and Sec. Computing*, 2006.