Jitter Analysis of Multimedia Streaming Traffic Assuming Batch Arrivals

Péter Zalán, Sándor Molnár and Tamás Éltető
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Budapest, Hungary
Email: zalan@tmit.bme.hu, molnar@tmit.bme.hu, eltetot@tmit.bme.hu

***Abstract*—This paper deals with the queueing analysis of IP networks carrying streaming traffic. The main focus of the analysis is the jitter which is a widely used performance characteristic of that type of traffic. The study is carried out by a novel technique using Quasi-Birth-Death processes in a matrix-geometric approach. Our method calculates network performance descriptors such as the transient queue length and the jitter characteristics in the case of correlated batch arrivals and Phase-type service times. The method is demonstrated by several numerical examples that are also compared to simulation results. A jitter measurement-based utilisation estimation method is also presented as an application of our results.**

I. Introduction

The mass distribution of multimedia content began in the first half of the 20^{th} century with the rise of the market of gramophone records. Later, the content broadcast became real-time thanks to the radio and television stations. At the beginning, the primary medium of the broadcast was the air. This situation changed when the cable tv became also popular particularly in cities where the high density of the potential audience made it commercially viable to deploy a wireline distribution network.

At the end of the 20^{th} century changes in the society and also a new medium, the Internet, generated new demands in the multimedia industry. One of them was the need for interactivity. Also, the technology development has made the operation and maintenance of IP-based networks cheaper and more amenable for multimedia content distribution. These and many other factors has been playing important roles in the current trends, that is, the multimedia content tends to be distributed using IP networks.

One advantage of IP-based multimedia content distribution by various streaming applications might be the possibility of the more efficient reuse of legacy networks, e.g. larger set of TV channels, invention of new services like video on demand, etc. However, the IP networks are packet-switched as opposed to the earlier circuit-switched media, therefore new network planning and design methods should be developed in order to maintain the accustomed level of QoS for the end-users.

In packet switched networks, queues are used for traffic multiplexing. The packets arriving at the same time to a multiplexing centre are queued and have to wait until they can be transmitted. This waiting time, the queueing delay, is an important matter of interest in the design of IP networks carrying multimedia traffic, because timing can be essential for the user perceived QoS. The one way or round-trip delay on a path is very important but the variation of the delay is even more important for the multimedia traffic. The reason is that the receiver should get the packets more-or-less continuously in order to be able to properly decode them and play the content without stops and distortions.

A popular descriptor of the variation in the packet delay is the jitter. The matter of our interest in this paper is the analysis of the transient queueing behaviour of multimedia traffic focusing on the estimation of the jitter as it is defined in [1] using numerical methods. One queue multiplexing a number of traffic flows is considered as it is shown in Figure 1. The multimedia traffic might have correlated arrival pattern, which has to be considered in the studies of its queueing behaviour. Here, a conservative approach was chosen to model the bursty packet arrivals using packet batches.

Besides this, another motivation for considering batch packet arrivals is, that it can be happen that the data link layer (e.g. ATM) splits the IP packets to several cells. This results in bursty cell arrival that we model as batch cell arrival process. Of course, in this case, the batch size distribution is strongly related to the IP packet sizes as it is shown in [2].

The batches are assumed to arrive according to the Poisson process in our paper. Though this assumption seems to be too restrictive, we note that we take it for the sake of simplicity. In fact, more general arrival processes (e.g. renewal processes, Markov Modulated Poisson Processes, Batch Markov Arrival Processes) could be handled as well. That is, this assumption is not critical.

The examples shown here consider batches of a few packets (i.e. 1–4). We note that this is not an artifact of the proposed approach as any batch size distribution (including distributions with unlimited support) can be considered. A limitation of the present analysis is that in case of general batch size distributions an approximation has to be applied. This approximation might change the results. This can be dealt with by improving the complexity and the quality of the approximation. The numerical properties of our method (e.g. approximation error, running time) heavily depends on the complexity of the applied model. Some practical examples are compared to simulations in order to evaluate the numerical errors of our method, but the deep assessment of the numerical properties is outside of

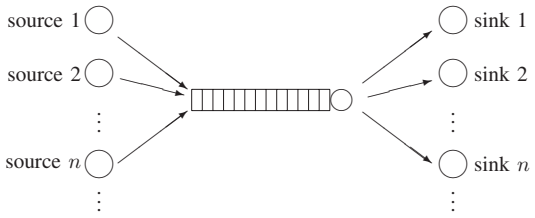

Fig. 1. Simple network model

the scope of this paper.

The paper also presents possible applications of the proposed method. The estimation of the link utilisation and/or the estimation of the available capacity is an important research topic. Using information on the buffer saturation level, it is possible to conclude about the link utilisation. However, measuring the number of the packets in a buffer is difficult while measuring the jitter is easier. We propose a method, which estimates the link utilisation based on jitter measurements as it is defined for RTP [1].

This paper is organised as follows. Section II discusses the related results for jitter analysis, queue length estimation (MBAC) and QBD. Section III contains the description of the method that calculates the transient distribution of a QBD. Generating functions and numerical integration are used calculating the distribution. Numerical examples and their practical motivation are shown in Section IV where the calculations and simulations are compared. In Section V, the potential applications are showed.

II. Related results

We have pointed out the significance of jitter from the viewpoint of multimedia traffic already in the Introduction. Nevertheless, the jitter is an important performance descriptor also for other reasons. For example, [3] investigates an ATM network with traffic policing. The jitter in this case is used to express the level of alteration that an initially periodic stream suffers due to the random queueing delays. A traffic policer should be designed so that these statistical variations do not cause unnecessary cell drops. We note, that our jitter definition is a bit different from the one in [3] as we defined the jitter according to RFC 1889, [1].

An important novelty in the present jitter analysis is the assumption of packet batches. Using Batch Markovian Arrival Processes (BMAP) the inherent correlations in the traffic can be better captured compared to other approaches. In [2], two traditional arrival models (Poisson process; MMPP) and a BMAP model were fitted to measurements and different performance descriptors were compared to each other and to the measurements. According to [2], the BMAP model generally performs better than the other models even when the queue length distributions developing in the different scenarios are compared.

A method estimating the link utilisation using jitter measurements will be proposed based on the presented numerical technique. Such link utilisation methods are important e.g. in the case of measurement-based admission control (MBAC) algorithms, where measurements provide input for the admission decisions. The most straightforward method might be to measure the number of connections as it is proposed in [4]. The task in this case is to use this information together with some knowledge on the statistical properties of the traffic (e.g. mean rate, peak rate or measured rate generating function) such that the admitted flows will experience the prescribed QoS. However, it is sometimes impossible because e.g. there is no entity that follows the number of connections in the network. What can be measured in this case is the queueing delay or the jitter. Of course, the estimations based on performance descriptors subject to certain variance cannot be exact. However, the uncertainty of the measurements can be incorporated into the MBAC algorithm as it is shown in [5] where such effects on the admission control are considered.

In this paper we present a numerical method and simulations to estimate the jitter according to RFC 1889 in a BMAP/PH/1 queueing system, where the term PH stands for Phase-type service times. The Phase-type distribution is a versatile class of probability distributions presented e.g. in [6], [7] that can efficiently be used in numerical methods. The tool used in our jitter analysis is based on the theory of quasi-birth-death (QBD) processes. For detailed description one can also refer to [6], [7]. Though a BMAP/PH/1 is not QBD in general, it is indeed a QBD assuming certain restrictions on the batch size distribution [8]. Furthermore, it is also shown in [8] that a BMAP/PH/1 queue can be approximated by an appropriate QBD.

In order to be able to estimate the jitter, the transient analysis of a BMAP/PH/1 queueing system is needed. Related results regarding the transient BMAP/G/1 queue can be found in [9]. Further results regarding the transient BMAP/PH/1 queue can be found in [10] and in an even more general setting in [11]. The latter algorithms have significantly better numerical properties compared to [9] because they heavily utilise the otherwise not too strong restriction taken on the service time distribution. Nevertheless, these algorithms still consider general batch size distributions, that might lead to numerical problems even in the computation of the stationary queue length distribution of a BMAP/PH/1 queue as it is shown in [8]. In order to be able to avoid these numerical difficulties, the method used in this paper uses the transient analysis of QBD processes.

Our proposed numerical method calculates the transient behaviour of discrete and continuous QBD processes using generating functions. We note, that numerical analysis of transient QBD processes for a type of jitter estimation has already been presented in [12]. However, there are important differences between our numerical technique and the one in [12]. First, a version of the folding algorithm was used in [12] that can be used for finite systems only. The approach used here can handle infinite systems as well. Second, BMAP arrivals are assumed in our work unlike in [12] where MMPP arrivals were assumed. We also note, that the jitter definition

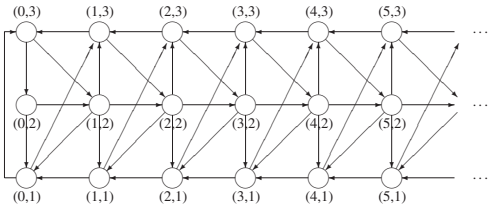

Fig. 2. An example for the QBD

in [12] is different from the one applied here. Our generating function technique is the generalisation of the one appearing in [7] for birth-death processes. According to the best of our knowledge, the technique shown in this paper has not yet appeared elsewhere.

The queueing system considered here has only one queue. When the majority of the delay on a path is gathered in a single bottleneck queue then the present results can directly be applied. Although the single bottleneck scenario is quite typical according to our practical experience, sometimes it might be important to be able to consider the jitter developing through the series of queues with significant queueing delays in each. Regarding this, one can find results in [13], [14] where tandem queues are analysed. These methods approximate the single server output with Markovian Arrival Process (MAP). Using these our jitter analysis can be extended to series of queues, that is the subject of future research and therefore out of scope of the present paper.

III. The Quasi-Birth-Death Process

In this section, first a short overview of the QBD processes is presented. Next, our technique calculating the transient distribution of a QBD is outlined. The calculations below are presented for discrete time QBD processes.

In the following, $\mathbf{P}$, $\mathbf{B}_0$, $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{R}(z)$, $\mathbf{S}(z)$, $\mathbf{R}$ and $\mathbf{S}$ denote matrices. $\vec{\mu}_0$, $\vec{\mu}_k$, $\vec{\mu}_{k,n}$, $\vec{X}_n(z)$, $\vec{p}(z)$, $\vec{q}(z)$, $\vec{t}(z)$ are vectors and z, k, m and n are scalars.

The quasi-birth-death process is a structured Markov process. Its state space consists of levels and every level contains phases. State transitions occur among phases within the levels or between two neighbouring levels only. The state transition matrix has a block tridiagonal form. The homogeneous QBD has the same state transition probabilities (rates) for one level to another, so the blocks in the three diagonal lines in (1) are the same: these blocks are $\mathbf{A}_0$, $\mathbf{A}_1$ and $\mathbf{A}_2$ the probability for the transitions up; within the level; down. The transition probabilities for the steps from the 0^{th} level to the 0^{th} are in $\mathbf{B}_0$.

$$\mathbf{P} = \begin{pmatrix} \mathbf{B}_0 & \mathbf{A}_0 & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} . \tag{1}$$

In Figure 2 one can see a QBD. Every level contains three phases, e.g. they are $(1,1)$, $(1,2)$ and $(1,3)$ in the first level. Transitions are allowed (e.g. from $(2,2)$) only within the level (to $(2,1)$ and $(2,3)$) or one level up (to $(3,2)$) or down (to $(1,1)$).

Let the vector $\vec{\mu}_k$ be the distribution of the Markov chain after k steps. $\vec{\mu}_k$ can be expressed with the transition probability matrix $\mathbf{P}$: $\vec{\mu}_k = \vec{\mu}_0\mathbf{P}^k$. Since the state space can be divided into levels, $\vec{\mu}_k$ can be divided into levels similarly. Let the vector $\vec{\mu}_{k,n}$ denote the probability vector in the n^{th} level after k steps. Using the special form of $\mathbf{P}$ we can write:

$$\vec{\mu}_{k+1,n} = \vec{\mu}_{k,n-1}\mathbf{A}_0 + \vec{\mu}_{k,n}\mathbf{A}_1 + \vec{\mu}_{k,n+1}\mathbf{A}_2 \ . \tag{2}$$

Let $\vec{X}_n(z)$ be the generating function for the n^{th} level:

$$\vec{X}_n(z) = \sum_{k=0}^{\infty} \vec{\mu}_{k,n} z^k \ .$$

According to (2) we have:

$$\vec{X}_n(z) = \vec{\mu}_{0,n} + \vec{X}_{n-1}(z)\mathbf{A}_0 z + \vec{X}_n(z)\mathbf{A}_1 z + \vec{X}_{n+1}(z)\mathbf{A}_2 z \ .$$

If the process starts from the l^{th} level then the solution has the following form:

$$\vec{X}_n(z) = \vec{p}(z)\mathbf{R}(z)^n + \vec{q}(z)\mathbf{S}(z)^{l-n} \ ,$$

if $0 \le n < l$, and

$$\vec{X}_n(z) = \vec{p}(z)\mathbf{R}(z)^n + \vec{q}(z)\mathbf{R}(z)^{n-l} \ , \tag{3}$$

if $n \ge l$, where $\mathbf{R}(z)$ and $\mathbf{S}(z)$ are the minimal nonnegative solutions of

$$\mathbf{R}(z) = \mathbf{A}_0 z + \mathbf{R}(z)\mathbf{A}_1 z + \mathbf{R}(z)^2\mathbf{A}_2 z \ ,$$
$$\mathbf{S}(z) = \mathbf{S}(z)^2\mathbf{A}_0 z + \mathbf{S}(z)\mathbf{A}_1 z + \mathbf{A}_2 z \ .$$

These matrix equations are solved using iterative substitutions similarly to the procedures shown in [7]. The justification of (3) can be found in [15].

(3) assumes infinite queue length, however, there are practical scenarios where the probability of saturation of a finite buffer is significant. In these cases the buffer length N should also be considered:

$$\vec{X}_n(z) = \vec{p}(z)\mathbf{R}(z)^n + \vec{q}(z)\mathbf{S}(z)^{l-n} \ + \vec{t}(z)\mathbf{S}(z)^{N-n} \ ,$$

if $0 \le n < l$, and

$$\vec{X}_n(z) = \vec{p}(z)\mathbf{R}(z)^n + \vec{q}(z)\mathbf{R}(z)^{n-l} \ + \vec{t}(z)\mathbf{S}(z)^{N-n} \tag{4}$$

if $l \le n \le N$.

The coefficients $\vec{p}(z), \vec{q}(z), \vec{t}(z)$ can be derived from the boundary (level 0, N) and from the initial (level l) conditions.

The probability of a level can be calculated from the generating function:

$$\vec{\mu}_{k,n} = \operatorname*{Res}_{z=0} \frac{\vec{X}_n(z)}{z^{k+1}} = \frac{1}{2\pi\mathrm{i}} \int_\gamma \frac{\vec{X}_n(z)}{z^{k+1}} \, \mathrm{d}z \ , \tag{5}$$

where γ is a closed curve around 0. The $\mathbf{R}(z)$ and $\mathbf{S}(z)$ matrix-valued functions are obtained by fixed point iterations and the (5) integral is approximated by summation. The vector-valued function $\vec{X}_n(z)$ has singularity at $z = 1$ since $\sum_{k=0}^{\infty} \vec{\mu}_{k,n}$ is divergent, therefore γ must not include 1.

The number of z points where the above calculations were evaluated depends on the parameters of the QBD and the requirements on the error control of the numerical calculations. The most straightforward error control is the calculation of the sum of all probabilities in the k^{th} step, that should be 1.

$$\sum_{n=0}^{\infty} \vec{\mu}_{k,n} \overrightarrow{\mathbb{1}} = 1,$$

where $\overrightarrow{\mathbb{1}}$ denotes the column vector of 1s. Note that, there is no need to evaluate $\vec{\mu}_{k,n}$ for each n. Instead, one can use the special structure of $\vec{X}_n(z)$ shown in (3):

$$\begin{aligned}\sum_{n=0}^{\infty} \vec{X}_n(z) \overrightarrow{\mathbb{1}} &= \vec{p}(z) \sum_{n=0}^{\infty} \mathbf{R}(z)^n \overrightarrow{\mathbb{1}} \\ + \vec{q}(z) \sum_{n=0}^{l} \mathbf{S}(z)^{l-n} \overrightarrow{\mathbb{1}} &+ \vec{q}(z) \sum_{n=l+1}^{\infty} \mathbf{R}(z)^{n-l} \overrightarrow{\mathbb{1}} \\ &= \vec{p}(z) \left(\mathbf{I} - \mathbf{R}(z)\right)^{-1} \overrightarrow{\mathbb{1}} \\ &+ \vec{q}(z) \left(\mathbf{I} - \mathbf{S}(z)^{l+1}\right) \left(\mathbf{I} - \mathbf{S}(z)\right)^{-1} \overrightarrow{\mathbb{1}} \\ &+ \vec{q}(z) \mathbf{R}(z)^1 \left(\mathbf{I} - \mathbf{R}(z)\right)^{-1} \overrightarrow{\mathbb{1}} \end{aligned} \quad (6)$$

where $\mathbf{I}$ denotes the identity matrix and $()^{-1}$ is the matrix inversion. There is a possible computational gain in the

$$\sum_{n=0}^{\infty} \mathbf{R}(z)^n = \left(\mathbf{I} - \mathbf{R}(z)\right)^{-1},$$

and

$$\sum_{n=0}^{l} \mathbf{R}(z)^n = \left(\mathbf{I} - \mathbf{R}(z)^{l+1}\right) \left(\mathbf{I} - \mathbf{R}(z)\right)^{-1}, \text{ etc.}$$

substitutions. Of course, these substitutions are possible only when the inverses exist, which is true in most cases according to our experience. If an inverse does not exist then a direct summation should be used instead. We note, that a similar expression can be derived for the summation of $\vec{X}_n(z)$ for finite queueing system in (4). Also, computational gains can be found for other summations involving $\vec{X}_n(z)$. For example, calculate the jitter as the following average of the absolute difference conditioning on the initial level (where $D_{0,k}$ denotes the change of the queue length after k steps, see the definition in (10)).

$$\mathbb{E}\left(|D_{0,k}| \,\middle|\, \text{queue length at 0 is } l\right) = \sum_{n=0}^{\infty} |n-l| \vec{\mu}_{k,n} \overrightarrow{\mathbb{1}}$$

Its generating function can be expressed by $\vec{X}_n(z)$ as

$$\begin{aligned}&\sum_{k=0}^{\infty} \mathbb{E}\left(|D_{0,k}| \,\middle|\, \text{queue length at 0 is } l\right) z^k \\ &= \sum_{k=0}^{\infty} \sum_{n=0}^{\infty} |n-l| \vec{\mu}_{k,n} \overrightarrow{\mathbb{1}} z^k = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} |n-l| \vec{\mu}_{k,n} \overrightarrow{\mathbb{1}} z^k \\ &= \sum_{n=0}^{\infty} |n-l| \vec{X}_n(z) \overrightarrow{\mathbb{1}} \\ &= \sum_{n=0}^{l} (l-n) \vec{X}_n(z) \overrightarrow{\mathbb{1}} + \sum_{n=l+1}^{\infty} (n-l) \vec{X}_n(z) \overrightarrow{\mathbb{1}}. \end{aligned} \quad (7)$$

The probability $\vec{\mu}_{k,n}$ can be calculated by applying a numerical approximation of (5), e.g.

$$\vec{\mu}_{k,n} \approx \sum_{m=1}^{M} \frac{\vec{X}_n(\gamma_m)}{\gamma_m^{k+1}} (\gamma_m - \gamma_{m-1}), \quad (8)$$

where γ_m is the m^{th} sample point of curve γ and $\gamma_0 = \gamma_M$. $\sum_{n=0}^{\infty} \vec{\mu}_{k,n} \overrightarrow{\mathbb{1}}$ can also be approximated numerically by substituting (6) in the place of $\vec{X}_n(z)$ in (8). This way one can evaluate the quality of the numerical approximation using the deviation from 1. If the deviation from 1 is not significant then the conditional average jitter $\mathbb{E}\left(|D_{0,k}| \,\middle|\, \text{queue length at 0 is } l\right)$ can be approximated by substituting (7) in the place of $\vec{X}_n(z)$ in (8).

The average jitter is obatined by deconditioning

$$\begin{aligned} J_k &= \sum_{l=0}^{\infty} \mathbb{E}\left(|D_{0,k}| \,\middle|\, \text{queue length at 0 is } l\right) \pi_l \\ &= \sum_{l=0}^{\infty} \sum_{n=0}^{\infty} |n-l| \vec{\mu}_{k,n} \overrightarrow{\mathbb{1}} \pi_l, \end{aligned} \quad (9)$$

where π_l, $l = 0, \ldots, \infty$ is the stationary distribution of the QBD that can be obtained using standard QBD solution techniques [6], [7].

We would like to note that numerical method presented in Section III is not sequential. The evaluation of the formulae can be done parallel for different complex γ_m values.

IV. Examples

The calculations presented in Section III consider discrete time QBD processes. Queueing systems can operate in discrete time (e.g. in ATM networks) or in continuous time (e.g. Ethernet networks). Since all examples are continuous time systems, the appropriate application of the uniformisation technique (also referred to as randomisation, for details see e.g. Section 2.8 in [16]) is needed where a continuous time Markov chain is translated to a discrete time Markov chain and vice-verse.

The numerical accuracy of the jitter estimation method presented in Section III depends on the complexity of the system to be analysed and also on the actual choice of parameters. Besides the jitter estimation itself, also an error control method was presented. We believe, that the possibility of such an error control is an important advantage over the simulation based estimation techniques in the following to situations.

1) It might happen that the parameters to be estimated (e.g. the jitter in our case) have large variance that decreases only slowly with time, particularly when the simulated parameter has strong correlations in time. In such case,

it is even difficult to determine, whether the parameter estimation converges or not.
In case of the jitter estimation of Section III these problems do not arise. If the system or the required parameter setup introduces numerical difficulties, then the error control method can give information on the accuracy. Moreover, the possibility of parallel processing can also reduce the running time of the calculations.

2) There are situations, when a parameter is interesting in the case of rare events. For example, we have to estimate the jitter in case, when the queue length is above a certain threshold. It is usually difficult to simulate rare events and estimate parameters in these periods.
If a rare event can be characterised e.g. as a subset of the state space in a Markovian model like ours, then the numerical analysis can be restricted to that particular subset regardless of the stationary probability of the subset.

Due to space limitations, we do not give a detailed numerical analysis of the jitter estimation method. Instead, we demonstrate the method in 3 queueing systems and 9 different parameter settings. The numerical calculations are compared to jitter estimations based on the simulations of the queueing models. The reason of this comparison is twofold. On one hand, our intention was to find some practical experience on the uncertainty of the parameter estimation from simulations. Therefore, not only the simulated averages, but the 5% and 95% quantiles of the simulated jitter values are plotted. On the other hand, a limitation of our approach is that there are cases where the queueing model is just an approximation of the "real system". In such cases, the numerical results are approximations only and the simulation results made it possible to evaluate the quality of the approximations.

The jitter definition in 6.3.1 of [1] is the following. *S_i is the RTP timestamp from packet i, and R_i is the time of arrival in RTP timestamp units for packet i, then for two packets i and j, D may be expressed as*

$$D_{i,j} = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i). \quad (10)$$

The interarrival jitter is calculated continuously as each data packet i is received from source SSRC$_n$, using this difference D for that packet and the previous packet $i-1$ in order of arrival (not necessarily in sequence), according to the formula

$$J = J + \frac{|D_{i-1,i}| - J}{16}.$$

From the above one can see, that the jitter depends on the timestamp unit. If the packet service time is fixed, for example 1 sec and the timestamp unit is also chosen to be the service time of a packet, then the jitter can be easily calculated using the queue length. Let L_n denote the queue length at the time $T_n = n\Delta T$, and $D_n = L_n - L_{n-1}$ is the change of the queue length. Then, the jitter is calculated at each monitoring as:

$$J_{n+1} = J_n + \frac{(|D_n| - J_n)}{16}.$$

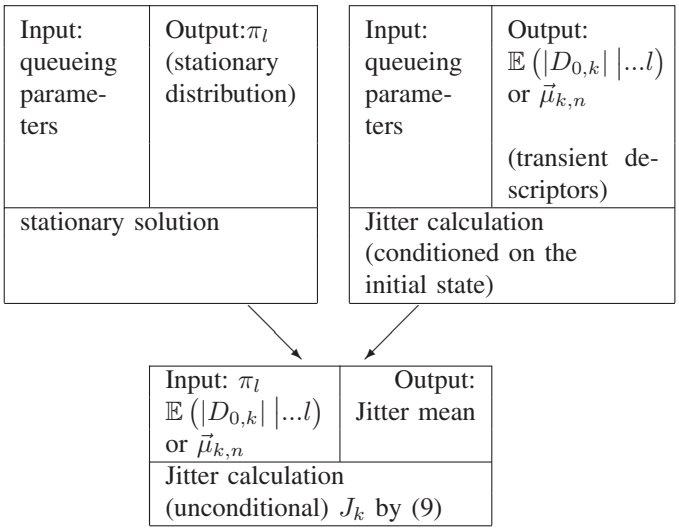

Fig. 3. Jitter calculation

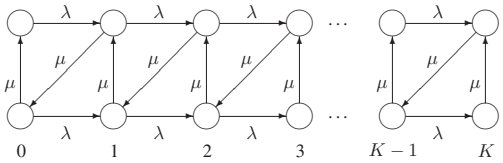

Fig. 4. State diagram of the $M/Erl_2/1/K$ system

This definition is used in the simulations. Of course, if the packet service time is not constant, but varies around one timestamp unit, then the above calculation becomes only an estimation of the jitter. We also note, that if the service time is a constant times one timestamp unit, then J_n should be multiplied by the same constant in order to get the jitter J according to [1].

The main steps of the average jitter estimation are shown in Figure 3. First, the stationary queue length distribution (π_l) is calculated using numerical techniques detailed e.g. in [6], [7]. The input parameters are the QBD parameters and the output is the stationary distribution.

Second, the conditional average of the absolute difference in the queue length after k steps in a discrete QBD process is calculated using (7) and (8). Alternatively, the whole transient distribution in the k^{th} step ($\vec{\mu}_{k,n}$) can be calculated and the average of the absolute difference is calculated using the transient distribution.

Third, the average jitter is estimated using (9), that is, the average absolute level (queue length) differences after k step starting from level l are weighted according to the stationary distribution. Alternatively, one can use the transient distribution ($\vec{\mu}_{k,n}$) here as it is indicated in Figure 3.

Three different queueing systems were analysed in order to show how our proposed jitter estimation method performs under various conditions.

A. The $M/Erl_2/1/K$ queue

The first system was the $M/Erl_2/1/K$ queue, that is, here the packets arrived according to a Poisson process. The

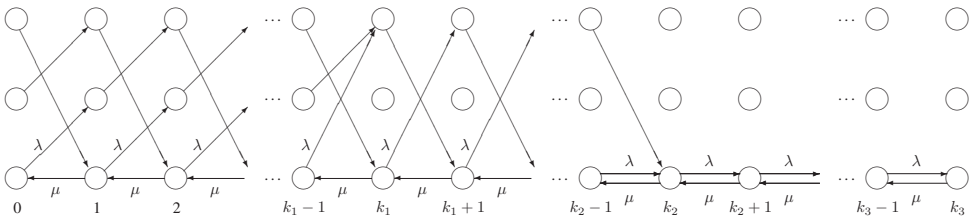

Fig. 5. State diagram of the $M^{[3]}/M/1/K$ system with RED-like packet discard

service time was a two-stage Erlang distribution. There was one server and finite queue length (40 packets) was assumed. This queueing system is intended to model the queueing in a bottleneck link for VoIP traffic. Though the pattern of the packets in a VoIP stream is usually regular (i.e. periodic), the Poisson arrival model here can be justified when there are many streams running parallel through the link. The fixed packet size is also quite typical in case of VoIP traffic. Since our jitter estimation method assumes Markovian service time, the Erlang distribution was used as an approximation because it has smaller coefficient of variation than the exponential distribution. The state transition diagram of the QBD model for this queue is shown in Figure 4.

B. The $M^{[3]}/M/1/K$ queue

The second system was the $M^{[3]}/M/1/K$ queue with a RED-like packet discard discipline (please refer to [17] for details about RED). Packet triples arrived according to a Poisson process and the queue length was finite ($K = 30$ packets). If the queue length was smaller than $K/3$ at the time of a batch arrival, then all three packets are admitted. If the queue length was between $K/3$ and $2K/3$ at an arrival then one packet was dropped and above $2K/3$ two packets were dropped from the packet triple. The motivation for this queueing system was a multiplexer where many video streams run parallel. The idea of using packet triples came from practical experience in a measurement of an MPEG stream, where three frames – one I, one B and one P frames – are sent close to each other and the next three frames arrived after a longer silent period. We note, that our investigations later revealed that this behaviour might not be typical since MPEG streams in different environments had different frame arrival pattern. Nevertheless, the arrival patterns were generally found to be correlated, therefore the batch arrival model of packet triples were chosen as a conservative approach. The packet discard discipline is based on the observation, that the MPEG frames are not equally important. The loss of B frames causes less damage than the loss of P frames and the most important ones are the I frames. Therefore, if a RED-like packet discard discipline is used then frames are dropped according to their importance: nothing; only one B frame; one B and one P frame. The state transition diagram corresponding to this queueing model is shown in Figure 5. The queueing system subject to numerical analysis was a QBD approximation of the $M^{[3]}/M/1/K$ according to [8].

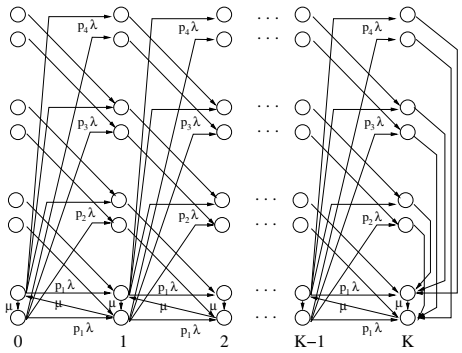

Fig. 6. State diagram of the $M^{[4]}/Erl_2/1/K$ system

C. The $M^{[4]}/Erl_2/1/K$ queue

The third queue was an $M^{[4]}/Erl_2/1/K$ system. The queue length was 40 packets. In this case the packets were discarded only in case when the queue was full. The batches in this case contained a random number of packets within 1 and 4 and the service time was a two-stage Erlang distribution. The choice of this queueing system was motivated by the case, when the data link layer is analysed where the segment size is smaller than the typical IP packet size, therefore the packets are split into several (at most 4) segments and these segments are transmitted in batches. Of course, the number of segments in a batch is determined by size of the IP packet transmitted in the corresponding batch. The state transition diagram of this model is shown in Figure 6. This queueing subject to numerical analysis was again a QBD approximation of $M^{[4]}/Erl_2/1/K$ according to [8].

D. Comparison of the jitter estimations with simulation results

Note, that a detailed investigation on the numerical accuracy of the jitter estimation method is out of scope of this paper. Nevertheless, we demonstrate the method using the above three queueing systems with link utilisations ranging from 0.1 to 0.9. The actual parameter choice does not have direct practical relevance since the main intention of the paper is merely to show that jitter measurements can be used to capacity estimation.

The average service times for all examples were chosen to be one 67 ms. The queue length was monitored in every 1.5 s. Both the numerical calculations and the simulations were performed using GNU Octave.

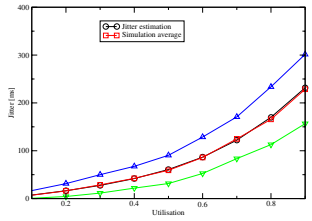

Fig. 7. Erlang service time ($M/Erl_2/1/K$ system)

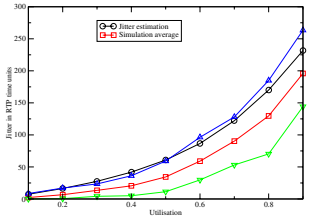

Fig. 8. Constant service time ($M/C/1/K$ system)

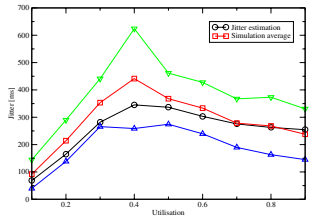

Fig. 9. Batch arrivals, exponential service time ($M^{[3]}/M/1/K$ system)

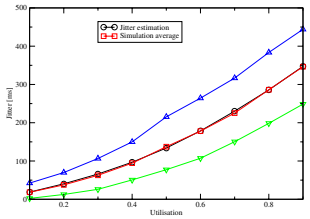

Fig. 10. Batch arrivals, Erlang service time ($M^{[4]}/Erl_2/1/K$ system)

The numerical results of the $M/Erl_2/1/K$ example and the simulations are shown in Figure 7. In this Figure and in the next Figures the triangles show the 5% and 95% quantiles of the jitter values in the simulations. One can see in Figure 7, that the calculated and simulated jitter averages match well for all link utilisations.

The choice of the Erlang service time in the $M/Erl_2/1/K$ example was because this system models a queue fed by VoIP flows, where the packet sizes are the same. Since the Erlang service time is not constant, the system is only an approximation of the real system. Figure 8 compares the jitter calculations to simulations where the service time was constant. As it might be expected, the constant service time has decreasing effect on the jitter that is confirmed by Figure 8. Our conclusion here is, that the two-stage Erlang distribution does not approximate a constant service time well in the present case. One solution is to apply higher order Erlang distributions, which would improve the quality of the estimations. The increased complexity of the QBD lead to increased calculation time and a tradeoff is needed between the model complexity and the expected quality of the estimations. However, the detailed analysis of the numerical properties of our proposed method is outside the scope of this paper.

The next queueing system is the $M^{[3]}/M/1/K$ queue with RED-like packet discard. The comparison results are shown in Figure 9. Again, the triangles show the 5% and 95% quantiles in the simulations.

One can see that the estimations of the jitter averages have the same behaviour as the simulated averages. That is, the jitter increases up to utilisation 0.4–0.5 and then it starts decreasing. It can also be seen in Figure 9 that for small link utilisations, the numerical method does not agree with the simulation mean, though the values remain in the 5%–95% interval. For utilisation levels above 0.7 the jitter estimations are much closer to the simulated averages. The observed differences for small link utilisations is due to numerical reasons that was also confirmed by the error control. According to the error control results, a better choice for the numerical parameters of the jitter estimation method improves its accuracy, therefore it is the subject of further research.

The last, and most complex example is the $M^{[4]}/Erl_2/1/K$ queue. The comparison results are shown in Figure 10. A single packet arrival occurred with 50% probability; the probability of a two packet batch was 30%; there were no three packet batches; the four packet batches occurred with 20% probability. As the error control had previously predicted during the calculations, the jitter estimations are quite accurate, the simulation averages well follow the calculations.

V. Application of the Results

The estimation of the utilisation of a bottleneck link is an important task. If there are means to access byte counters in the interface e.g. via SNMP queries or even if a monitoring system like MRTG is installed then the situation is quite simple. The network providers, however, are not generally willing to allow the users or content providers such access, therefore it is necessary to find alternative methods. There are bandwidth estimation methodologies using active probing between end hosts. Most of these methods are designed so that the probing has negligible disturbing effects on the network traffic and the traffic volume of several links can be measured at the same time.

One important aspect of the active probing techniques is that they use high precision hardware equipment and certain level of cooperation between two or more end hosts. This is not a significant issue from O&M perspective, but as it was pointed out in the Introduction, the main interest of this paper is related to the IP-based multimedia content distribution. In this case, neither high precision hardware at the clients nor significant cooperation between them can be expected as this latter might involve difficult system design task and even legal issues.

Therefore, simple passive monitoring methods are preferred in such systems like delay or jitter measurements as it is proposed in the RTP protocol in RFC 1889 in [1]. Based on the jitter estimation method presented in this paper we propose a simple two step link utilisation estimation procedure shown in Figure 11. The procedure assumes that the relation between the measured jitter and the link utilisation like Figure 7–10 is already known[1]. The first step of the procedure is the measurement of the jitter and the packet loss using RFC 1889 during a period of time (e.g. 10 sec). Second, the average jitter is compared to the table. It might be, that the link is overloaded, that is, the utilisation is above 100%, in which case the jitter can be small again because the buffer is full

[1] In the case of the $M^{[3]}/M/1/K$ system, the relation between the jitter and the utilisation is not monotone. This difficulty can be handled by estimating e.g. the variance of the jitter besides the average. This can also be done with a simple extension of the numerical method. The jitter average and variance together completely characterise the utilisation for the $M^{[3]}/M/1/K$ system.

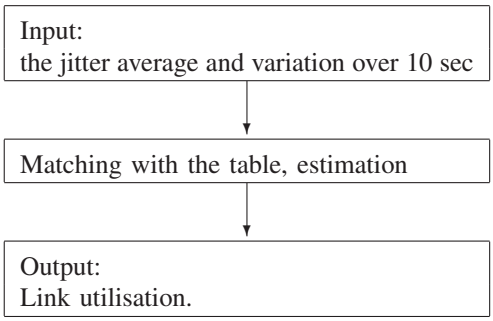

Fig. 11. Determination of the utilisation

most of the time. However, in this case, there is significant packet loss, so the clients can conclude about the heavy traffic conditions. Finally, given that there is no significant packet loss, the link utilisation is estimated.

The link utilisation estimation procedure can be used by the clients to inform the content provider about the network conditions. In this method we assume that the jitter estimation calculated by clients is based on time average values. This way, the content provider has information about the increasing network traffic before the QoS of the multimedia streams start to degrade. Furthermore, the above procedure can be used for measurement-based admission control.

As long as the call admission decision is made by a server which is aware of the used and available resources the CAC method is straightforward: if there is enough available bandwidth for a new connection then it is admitted. An important disadvantage here is related to the configuration of the CAC, since it should be aware of the available capacities in its domain. In a measurement-based admission control, the available link capacity is measured and the system can more easily accommodate itself to the changes in the network. Indeed, in [18] a distributed call admission control is proposed where the decisions are done by the clients using capacity estimation based on the measurement of "control transmissions". A similar distributed CAC can be developed using our proposed capacity estimation procedure for multimedia traffic in IP networks. For example, when a client wishes to establish a new streaming flow, a short measurement period provides estimation on the link utilisation using the jitter. If the estimation is within a prescribed bound then the flow can be started.

VI. Conclusions

A numerical method estimating the jitter in a bottleneck queue assuming batch arrivals and Erlang service time is presented in this paper. The numerical method is based on transient QBD analysis. The method was illustrated using three queueing systems motivated by practical scenarios. The service time was fixed in the examples and the arrival rates were changed in order to estimate the jitter developing under different link utilisations. The results are also compared to simulations.

A link utilisation estimation procedure as a possible application of the numerical method was also presented. A system using measurement-based admission control based on existing ideas was also outlined.

Though the jitter estimation method currently considers a single queue, the extension of the method to tandem queues is possible using existing results. The proposed application of the jitter estimation method might make id possible to establish various MBAC systems for the load control of IP-based multimedia content distribution. Further research in this direction is also planned.

References

[1] A.-V. T. W. Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 1889, Jan. 1996.

[2] A. Klemm, C. Lindemann, and M. Lohmann, "Modeling IP traffic using the batch Markovian arrival process." *Perform. Eval.*, vol. 54, no. 2, pp. 149–173, 2003.

[3] J. Roberts and F. Guillemin, "Jitter in ATM networks and its impact on peak rate enforcement." *Perform. Eval.*, vol. 16, no. 1-3, pp. 35–48, 1992.

[4] R. J. Gibbens and F. P. Kelly, "Measurement-based connection admission control," in Proc. 15th International Teletraffic Congress, Jun. 1997.

[5] M. Grossglauser and D. N. C. Tse, "A framework for robust measurement-based admission control." *IEEE/ACM Trans. Netw.*, vol. 7, no. 3, pp. 293–309, 1999.

[6] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach*. The Johns Hopkins University Press, 1981.

[7] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ser. ASA-SIAM Series on statistics and applied probability. Philadelphia PA: SIAM, 1999.

[8] T. Éltető and M. Telek, "Numerical analysis of M/G/1 type queueing systems with phase type transition structure," *Journal of Computational and Applied Mathematics*, 2006.

[9] D. M. Lucantoni, G. L. Choudhury, and W. Whitt, "The transient BMAP/G/1 queue," *Stochastic Models*, vol. 10, no. 1, pp. 145–182, 1994.

[10] L.-M. L. Ny and B. Sericola, "Transient analysis of the bmap/ph/1 queue," *International Journal of Simulation: Systems, Science and Technology. Special Issue on Modelling and Simulation of Computer Systems*, vol. 3, no. 3, pp. 4–14, 2003.

[11] H. Masuyama and T. Takine, "Algorithmic computation of the time-dependent solution of structured markov chains and its application to queues," *Stochastic Models*, vol. 21, no. 4, pp. 885–912, 2005.

[12] S. Q. Li and H. Sheng, "A generalized folding algorithm for transient analysis of finite QBD processes and its queueing applications," in Computations with Markov Chains, Proc. of the 2nd Int. Workshop on the Numerical Solution of Markov Chains, pp. 463–482, 1995.

[13] A. Heindl and M. Telek, "Output models of MAP/PH/1(/K) queues for an efficient network decomposition." *Perform. Eval.*, vol. 49, no. 1/4, pp. 321–339, 2002.

[14] ——, "MAP-based decomposition of tandem networks of ./PH/1(/K) queues with MAP input." in *MMB*, 2001, pp. 179–194.

[15] P. Zalán and T. Éltető, "Transient analysis of QBD processes," Available from the authors.

[16] V. Ramaswami and G. Latouche, "A general class of Markov processes with explicit matrix-geometric solutions," *Operation Research Spectrum*, vol. 8, pp. 209–218, 1986.

[17] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, 1993.

[18] M. Xiao, N. B. Shroff, and E. K. P. Chong, "Distributed admission control for power-controlled cellular wireless systems," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 790–800, 2001.