

University of Würzburg  
Institute of Computer Science  
Research Report Series

**An Analytical Model for the Evaluation  
of the Information Processing in  
Multi-Probe Networks**

Thomas Zinner<sup>1</sup>, Dirk Staehle<sup>1</sup>, Simon Oechsner<sup>1</sup>, Phuoc  
Tran-Gia<sup>1</sup> and Kurt Tutschku<sup>2</sup>

Report No. 456

March 2009

<sup>1</sup> University of Würzburg  
Institute of Computer Science  
Department of Distributed Systems  
Am Hubland, D-97074 Würzburg, Germany  
{zinner,dstaehle,oechsner,trangia}@informatik.uni-wuerzburg.de

<sup>2</sup> University of Wien  
Institute of Computer Science  
Department of Future Communication  
Universitätsstrasse 10, 1090 Wien, Austria  
kurt.tutschku@univie.ac.at



# An Analytical Model for the Evaluation of the Information Processing in Multi-Probe Networks

**Thomas Zinner, Dirk Staehle,  
Simon Oechsner, Phuoc Tran-Gia**  
University of Würzburg  
Institute of Computer Science  
Department of Distributed Systems  
Am Hubland, D-97074 Würzburg,  
Germany  
{zinner,dstaehle,oechsner,trangia}  
@informatik.uni-wuerzburg.de

**Kurt Tutschku**  
University of Wien  
Institute of Computer Science  
Department of Future Communication  
Universitätsstrasse 10, 1090 Wien, Austria  
kurt.tutschku@univie.ac.at

## Abstract

For the Future Internet and the Internet of Things, access and transport networks with much higher capacity than today are expected. As a result the bottleneck in large scale multi-probe networks will move from the network to the processing units, denoted as *information sinks*. Our contribution is a generic analytical performance model which enables an analysis of the resource requirements and handling at these information sinks. We will consider the required amount of buffer space at the information sink for a certain target duration until all requests are processed within an interval. This duration is denoted as reaction time. The presented model and the performed investigations can support application designers to choose the right mechanisms required for their scenario.

## 1 Introduction

Multimedia monitoring and surveillance are expected to become major applications of the Future Internet (FI) [1] and the Internet of Things (IoT) [2]. This is witnessed by initiatives like the European FP7 work program and technical projects such as *Urban Sensing* [3], *Microgrid* [4, 5], *Wireless Camera Arrays* [6], and national security infrastructures [7, 8].

Current networked applications are designed for meeting the technical constraints of networks. Networked systems, for example probes, are located in Local Area Network (LANs) which are connected by access gateways to the Internet, cf. Figure 1. The transmission capacity in a LAN is typically very high ( $\sim 100$ - $1000$  Mbps) while the capacity of access links and the per-flow throughput in the transport networks is typically much lower ( $\sim 1$ - $10$  Mbps). Hence, access link and transport network constitute a bottleneck from application perspective [9]. As a result, the designers of distributed applications have to adapt their applications to this impasse. The adaption can be achieved, for example, by reducing and shaping the traffic which is injected into the network.

The Future Internet might reverse this approach for *networked application design*. The FI may provide very high capacities on access links and in transport networks. For example, the anticipated uplink capacity on advanced LTE (Long Term Evolution) connections in future 4G wireless system is about 10-100 Mbps [10] and 100Gbps Ethernet is

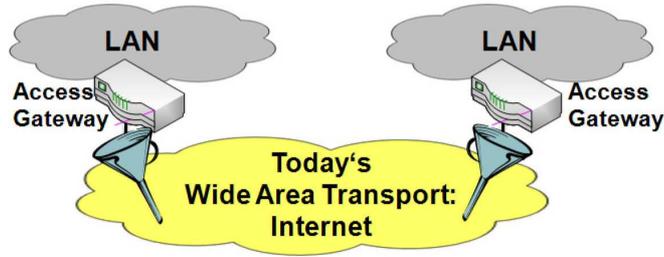


Figure 1: Today's Bottlenecks

currently investigated as a core network transport system [11]. This increase of capacity might decrease the characteristic of the network as a whole of being the bottleneck, cf. Figure 2.

Moreover, these changes in capacity, together with the concept of *Network Virtualization* [12,13], will allow for an evolution in system design: network engineers should *design networks for applications* rather than application developers designing applications for networks. As result for application designers, the FI will require them to reconsider their system architecture. In particular new bottlenecks will appear at application components such as servers or data processing units, in general denoted as *information sinks*. Especially, in the context of the IoT where multimedia monitoring and surveillance will be done by a huge number, even millions, of probes. It is expected that severe bottlenecks will occur at these information sinks.

In this paper we will investigate how to design the information sink in large scale multi-probe networks under the assumption that data transmission in the network does not constitute a bottleneck. Particular focus is laid on the definition of a generic performance model since specific applications can't yet be foreseen, the computability of the model, and the discussion of the trade-off between reaction time and memory. Reaction time in this context is defined as the duration needed for processing the overall amount of data from all probes within an interval. This investigation will support application designers to choose at a very early stage in the design process the right mechanism required for

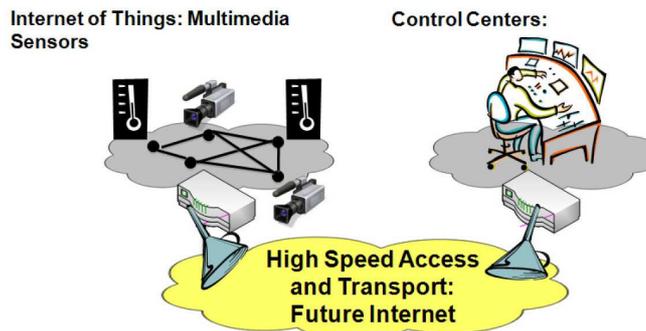


Figure 2: Future Bottlenecks

their scenario. In detail, the investigation will guide them whether to increase queuing memory or processor speed in a specific scenario.

The paper is structured as follows: Section 2 discusses the related work done in this field of research. Section 3 details the generic application scenario. Section 4 introduces the considered performance model in detail and Section 5 presents the results of the investigations. Finally, Section 6 will conclude the paper and gives an outlook for further research.

## 2 Background and Related Work

In [6] the authors describe a bandwidth management framework for wireless camera array. They address the problem of bandwidth management in order to coordinate multiple video flows in order to support wireless video streaming. For that different relations and scheduling policies are investigated. In contrast to our contribution, which assumes the information sink to be the bottleneck in the system, the authors assume the wireless channel to be the bottleneck.

In [14] an analytical framework based on fluid models for large-scale wireless sensor networks is developed. The approach discusses in detail energy consumption, channel contention and traffic routing. Compared to [14], our approach focuses on the information sink and not on the sensor network itself.

There are many queuing theoretical approaches for investigating server systems. In [15] the buffer requirements for an ATM multiplexer were evaluated. Although there are similarities to our system, the presented model in [15] differs from our since we investigate different sizes of probe types which transmit their data in periodical intervals. Furthermore, we investigate the scheduling of the transmissions within such an interval.

We assumed in [8] that all probes generate data following one single probability distribution. In this paper we present is an extension of the model in [8]. We will consider multiple types of probes which generate data and send it the information sink according to different distributions.

## 3 System Description

The general kind of system we consider in this paper consists of one central server that periodically receives information from a number of sources connected to this server via a network. This information is processed by the server, with information that is received while the server is busy being buffered. Each source may send a random amount of information per time interval, which is also termed time slice.

Examples for this kind of architecture are a server that gathers and processes pictures from surveillance cameras in periodic intervals, or network probes monitoring certain traffic characteristics and reporting their findings in regular intervals back to a network management or anomaly detection system.

In the latter example, the information consists of counters updated by the probes in each interval, with each probe being able to report different numbers and types of counters. In current systems, general values logged by SNMP may be reported, such as

the number of packets going over a link, as well as more specific characteristics, such as the amount of traffic from a certain range of IP addresses. The server updates its current view on the network in each interval and has to process all reported counter values in an interval in order to do so. It may process single counters at a time and buffer the rest (from the same probe or from others) in a queue. The processing of a counter may include a comparison with older similar values, checking for violated thresholds, or simply storing it in a database.

One aim of such a server is generating a fast new overview on the system state, i.e., a fast processing of the complete set of data generated by all sources in one time slice. This has to be weighed against the buffer space needed to store data that could not yet be processed. This would not be a problem if the server simply polled the sources one after another, having only to store the data of one source at a time. However, in general the communication between the sources and the server follows the push-model in the examples described. The sources are therefore not polled by the server, but send their data self-triggered. The only way the sending behavior of a source is influenced is by a rough setting of the time it sends its data in relation to the other sources.

In the following, we will describe an analytical model of this family of systems and provide insights in how the sending behavior of the sources may be influenced in order to adapt the server load as necessary for given resources.

## 4 Model

In this Section, we describe the queuing model used to analyze the system as described above.

### 4.1 Abstract Server Model and Performance Metrics

We consider scenarios with multiple clients and one server. For instance, clients can be network or surveillance probes which send a packet consisting of a specific amount of data to the server. This data is modeled as work to be done by the server, e.g., it stores the data in a database and performs a computation upon this data. Data arriving at the server while it is busy is buffered into a queue, which we assume to be infinite throughout this work. This assumption simplifies the analysis of the system and allows us to discuss overall processing times and buffer sizes. We further assume the amount, and therefore the processing time, of data per packet follows an arbitrary distribution. Due to the diverse placement of the probes participating in such a system, this distribution is basically different for each probe. Nevertheless, there may be probes with similar processing time distributions. Thus, these probes approximately follow a common distribution which simplifies the analysis. The probes transmit their data in equally spaced time slices, with a constant length  $\tau$ . As performance metrics for the system we define the buffer occupancy  $O$  after the last arrival within a time slice and the reaction time  $R$  as the time needed until the last request within a time slice is processed. We will identify the trade-off between these two parameters. It should be noted, that, especially for early detection systems, an interrelated analysis of all data

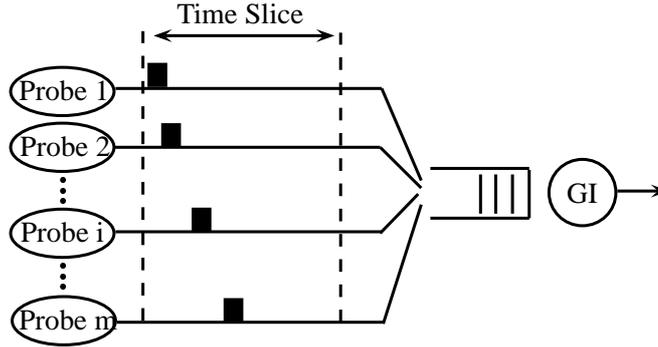


Figure 3: Queuing model

transmitted within a time slice may be required. Thus, an objective may be for instance to minimize the buffer occupancy for a given reaction time.

## 4.2 Performance Model

We model the described system as a  $D/GI/1 - \infty$  delay queuing system as shown in Fig. 3. We assume an infinite queue with a first-come-first-served order, i.e. no probe data will be lost.

There are  $n$  probes in the system which send their data during a time slice to the server. The state space is defined by means of the unfinished work in the system,  $U$ . The amount of unfinished work generated by a probe at the server follows a probability distribution  $b_i(t)$  with mean  $E[B_i]$  and standard deviation  $STD[B_i]$ . There exist  $K$  different distributions  $b_K(t)$  with  $1 \leq K \leq n$ , i.e.  $K$  different types of probes. In case of  $K = n$  all distributions are different and can not be combined to several types, i.e. the analysis can not be simplified. For  $K = 1$  the analysis presented in [8] is sufficient to produce the same results.

The total number of probes following distribution  $b_i(t)$  is  $n_i$ ,  $1 \leq i \leq K$ , and the total number of probes is  $n = \sum_{i=1}^K n_i$ . The number of probes  $m$  which have transmitted their data within a time slice is incremented after each new arrival. According to the type of the probe distribution  $b_i(t)$  at this arrival, the corresponding counter  $m_i$  is also increased. Thus we define the state vector  $\bar{m}$  denoting for each arrival how many probes of each type have already transmitted their data, i.e.  $\bar{m} = \{m_1, m_2, \dots, m_K\}$ .

The average amount of unfinished work  $E[B]$  added by a probe to the system is defined as  $E[B] = \sum_{i=1}^m \frac{n_i}{n} \cdot E[B_i]$

We compute the desired system parameters buffer occupancy  $O$  and the reaction time  $R$  with a discrete-time analysis [16] [17].

## 4.3 Arrival Patterns

We consider the arrival pattern *Distributed arrival pattern with processing phase* which was introduced in [8]. This pattern describes how the network probes are scheduled to

send their measurements.

Considering this arrival pattern the time slice  $\tau$  is divided into two intervals  $\tau'$  and  $\tau - \tau'$ , which we denote as *transmission phase* and *processing phase*, respectively. This approach is shown in Fig. 4. In the transmission phase  $\tau'$ , all probes are sending their data to the server with constant inter-arrival times  $\tau'/m$ . The order in which the probes transmit their data can change from time slice to time slice. The processing phase  $\tau - \tau'$  is then exclusively used for processing the data. For the case  $\tau' = 0$ , which we call *Super Batch Arrival*, all probes transmit their data simultaneously. Due to the cumulated arrivals this leads to increased demands on the buffer capacity of the information sink. The case  $\tau = \tau'$  we denote as *Distributed Arrival*. Here the arrivals are spread equally over the whole time slice, which leads to a lower maximum utilization of the network as well as of the buffer occupancy. However, idle times at the server appear between two arrivals, which increase the processing time of a whole time slice. For  $\tau' < \tau$  the idle time between two probe arrivals can be decreased. Furthermore, for  $\tau' > 0$  the buffer requirements are reduced compared to the case  $\tau' = 0$ . We will see later how the parameter  $\tau'$  can be used to tune the system either for buffer efficiency or for shorter processing times, and thus for faster reactions.

It should be noted, that additional probes can be easily added to an existing installation by scheduling them after the last probe arrival. In this case, the length of the transmission phase changes.

#### 4.4 State Transitions and Computation of the Performance Metrics

We define the beginning of each time slice as observation points, where we investigate the unfinished work  $u(t)$ . These points are regeneration points, i.e., all of the past history that is pertinent to future behavior is completely summarized in the current value of  $u(t)$ . Transitions between different states are described with state transition probability matrices. A diagram of the investigated process including the state transition probability matrices is depicted in Figure 5. We assume an equally spaced inter-arrival time  $\Delta t$  between the requests in a time slice. Depending on the arrival pattern, the inter-arrival time takes values  $\Delta t \in [0, \frac{\tau}{m}]$ . We will use the following notation, which is also used in Figure 5:

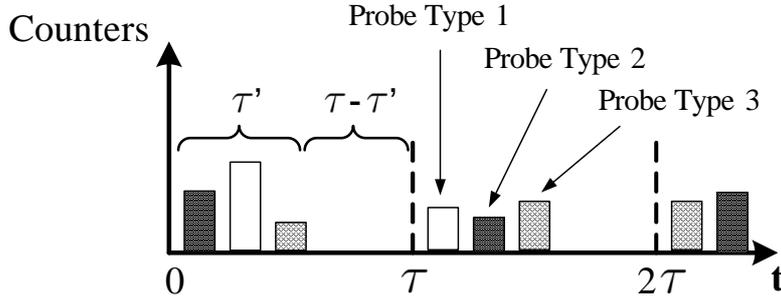


Figure 4: Distributed arrivals with processing phase

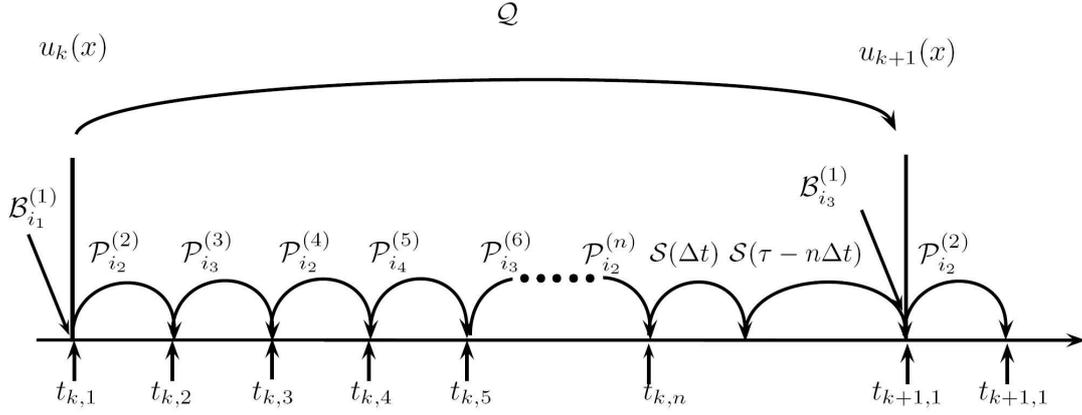


Figure 5: Time diagram of the process including state transition probability matrices

- $u_k(t)$  : unfinished work at the begin of time slice  $k$
- $t_{k,l}$  : time of the  $l$ -th arrival in time slice  $k$
- $\mathcal{Q}(\tau)$  : state transition probability matrix between two observation points
- $\mathcal{B}_i$  : state transition probability matrix increasing unfinished work by  $b_i(t)$
- $\mathcal{P}_i(\Delta t)$  : state transition probability matrix between two arrivals increasing unfinished work by  $b_i(t)$  and decreasing the unfinished work by  $s(\Delta t)$
- $\mathcal{S}(\Delta t)$  : state transition probability matrix depicting a reduction of unfinished work for an interval

The state transition probability matrix  $\mathcal{P}_i(\Delta t)$  describes the processing of unfinished work between two successive arrivals and the unfinished work added by a new arrival. The processing, depending on the inter arrival time  $\Delta t$ , is denoted by the probability matrix  $\mathcal{S}(\Delta t)$ . According to the probe type  $i$  of the current arrival, the amount of unfinished work added to the server is represented by the probability distribution  $b_i(t)$ , equivalent to the transition probability matrix  $\mathcal{B}_i$ . Thus  $\mathcal{P}_i$  is computed as:

$$\mathcal{P}_i = \mathcal{S}(\Delta t) \cdot \mathcal{B}_i \quad (1)$$

with

$$\mathcal{B}_{i,jk} = p(B_i = k - j), \quad (2)$$

$$\mathcal{S}_{jk}(\Delta t) = \begin{cases} 1, & \text{if } k = \max\{0, j - \Delta t\} \\ 0, & \text{else.} \end{cases} \quad (3)$$

At the regeneration points the remaining work load  $U_k$  can be computed, cf. [18], as

$$u_k(t) = u_{k-1}(t) \cdot \mathcal{Q}. \quad (4)$$

In case of a stationary system, i.e  $\rho < 1$ , the steady state equation is

$$u(t) = u(t) \cdot \mathcal{Q}. \quad (5)$$

Thus, the steady state distribution at the observation points can be computed by finding the left eigenvector for the probability transition matrix  $\mathcal{Q}$ . Then, we can compute the steady state distribution  $o_l(t)$  at the time of arrival  $l$  by

$$o_l(t) = u(t) \cdot \mathcal{B}_{i_1}^1 \cdot \mathcal{P}_{i_2}^2 \dots \cdot \mathcal{P}_{i_k}^l \quad (6)$$

$$= u(t) \cdot \mathcal{Q}_l, \quad (7)$$

whereas  $\mathcal{Q}_l$  expresses the transition path to the investigated arrival.

The reaction time  $R$  is the sum of the service time  $B_m$  of the last request within a time slice and the unfinished work  $O_m$ , queued in the buffer at the arrival time of this request, i.e  $R = (n - 1) \cdot \Delta t + O_n + B_n$ . Therefore we compute the reaction time distribution as

$$r(t) = \delta(t - ((n - 1)\Delta t)) \cdot o_n(t) \cdot b_n(t) \quad (8)$$

But before we can compute the steady state distribution of unfinished work and the buffer occupancy at the arrival times, we have to compute the transition probability matrix  $\mathcal{Q}$ . The algorithm for that is described in the following.

Let us observe an random arrival time in an arbitrary time slice, where  $m$  probes have already transmitted their data. The state vector  $\bar{m}$  contains the number of the different probes which have transmitted their data,  $\bar{m} = \{m_1, m_2, \dots, m_K\}$ . Since at each arrival time one probe transmits its data, there exist at most  $K$  different predecessor states  $\bar{m}_k^{-1}$ ,  $1 \leq k \leq K$  for that state. Here  $\bar{m}_k^{-1}$  denotes the predecessor state  $\bar{m}_k^{-1} = \{m_1, m_2, \dots, m_{k-1}, m_k - 1, m_{k+1}, \dots, m_K\}$  and an arrival of unfinished work denoted by  $\mathcal{B}_k$ . Thus, we define the set  $\bar{k}$  denoting the possible predecessor states as

$$\bar{k} = k : \exists \bar{m}_k^{-1} | 0 < m_k \leq n_k. \quad (9)$$

The state transition probability matrix to an arbitrary arrival  $\mathcal{Q}_{m, \bar{m}}$  is computed as

$$\mathcal{Q}_{m, \bar{m}} = \sum_{k \in \bar{k}} \frac{n_k - (m_k - 1)}{n - (m - 1)} \cdot \mathcal{Q}_{m-1, \bar{m}_k^{-1}} \cdot \mathcal{P}_k(\Delta t),$$

$$\mathcal{Q}_{1, \bar{m}} = \sum_{k \in \bar{k}} \frac{n_k}{n} \cdot \mathcal{Q}_{0, \bar{m}_k^{-1}} \cdot \mathcal{B}_k(\Delta t),$$

$$\mathcal{Q}_{0, \bar{m}} = \mathcal{I}$$

An exemplary state transition is depicted in Fig. 6. There are  $k$  possible predecessor states  $\bar{m}^{-1}$  which may lead to the current state  $\bar{m}$ . Thus the transition probability matrix for state  $\mathcal{Q}_{m, \bar{m}}$  can be computed from these states.

With this result we easily can compute the transition probability matrix between the observation points as

$$\mathcal{Q} = \mathcal{Q}_{n, \bar{m}} \cdot S(\Delta t) \cdot S(\tau - n \cdot \Delta t). \quad (10)$$

With this result we first can compute the steady state distribution of the unfinished work at the observation points, i.e. at the beginning of each time slice, and from this the reaction time  $R$  and the buffer occupancy  $O$ .

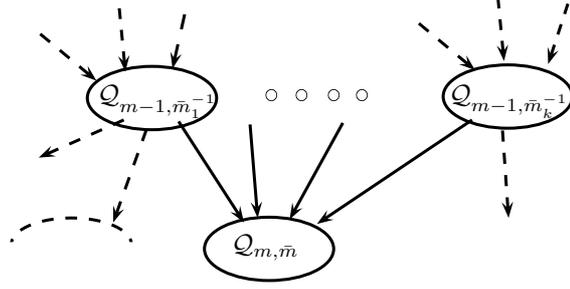


Figure 6: n-dimensional

#### 4.5 Example for $K=2$

In this subsection we give an example for two different probe types, i.e.  $K = 2$ . The total number of probes participating in the system  $n = 3$ , the number of probes of the first type  $n_1 = 2$ , and the number of probes of the second type  $n_2 = 1$ . The unfinished work added to the system by an arrival of type 1 follows the distribution  $b_1(t)$ , by an arrival of type 2 distribution  $b_2(t)$ , respectively. For given  $\tau$  and  $\tau'$ , the transition probability matrices between two successive arrivals,  $\mathcal{P}_1$  and  $\mathcal{P}_2$  can be computed. With these matrices, the different possible state transition probability matrices can be computed, as depicted in Fig. 7. This figure illustrates the advantage of our algorithm to compute the state transition probability matrix between the observation points. Let us regard the transition state  $Q_{2,(1,1)}$ . This state combines the paths in which one probe of each type has transmitted their data within the current time slice. Both possible transmission sequences are weighted with their probability of occurrence into this transition state. The next transition state  $Q_{3,(2,1)}$  can be computed on the basis of its direct predecessor states and does not have to rely on the sequence of the transmissions.

### 5 Numerical Results

We will now present results from this analysis that give a first impression of the system performance. We consider heterogeneous scenarios with two different probe types, i.e.

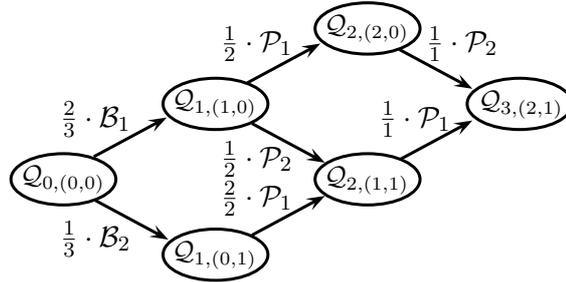


Figure 7: Two-dimensional state transition diagram

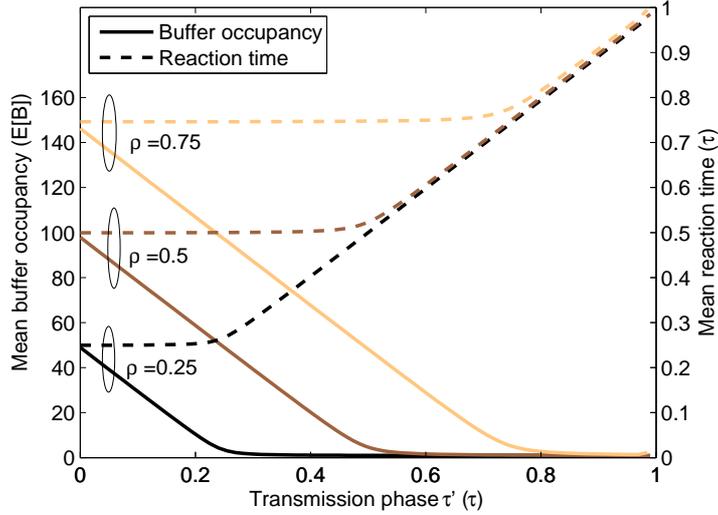


Figure 8: Mean buffer occupancy and mean reaction time vs.  $\tau'$

probes transmit unfinished work according to two different distributions  $b_1(t)$  and  $b_2(t)$ . If not stated otherwise the buffer occupancy is given as multiples of the average amount of work  $E[B]$  generated by a probe. The reaction time, as well as the length of the transmission phase  $\tau'$ , is presented as fraction of the time slice length  $\tau$ .

### 5.1 Macroscopic behavior

In a first investigation we assume these distributions to follow negative binomial distributions with  $E[B_1] = \frac{25}{10000} \cdot \tau$  and  $E[B_2] = \frac{75}{10000} \cdot \tau$ . As standard deviation we assume  $STD[B_1] = STD[B_2] = \frac{20}{10000} \cdot \tau$ , which leads to coefficient of variations  $c_{v1} = 0.8$  and  $c_{v2} = 0.27$ . Thus, we have modeled one source type with a highly variable output, and one type that varies only little in the amount of work generated. We now examine the influence of the length of the transmission phase  $\tau'$  on the mean buffer occupancy and on the mean reaction time as shown in Fig.8. We assume different load values as  $\rho \in \{0.25, 0.5, 0.75\}$  with a corresponding number of probes  $n \in \{50, 100, 150\}$ , whereas  $n_1 = n_2$ . The solid lines indicate the average buffer occupancy and the dashed lines show the corresponding reaction time, i.e. the time until the last probe data has been processed.

Let us define  $\tau'^* = \rho \cdot \tau$  as average time the sink needs to process all data *with no* idle phases occurring. Thus, the curves are divided into two parts, one with  $\tau' < \tau'^*$  and one with  $\tau' > \tau'^*$ .

For the case  $\tau' < \tau'^*$ , the arriving measurement data can not be entirely processed in the transmission phase, queuing occurs and the buffer occupancy increases with decreasing  $\tau'$ . For  $\tau' = 0$  the arrivals correspond to the super batch arrival scheme, i.e. the measurement data of all probes arrive simultaneously at the information sink and the buffer occupancy reaches its maximum. Accordingly, the mean reaction times correspond

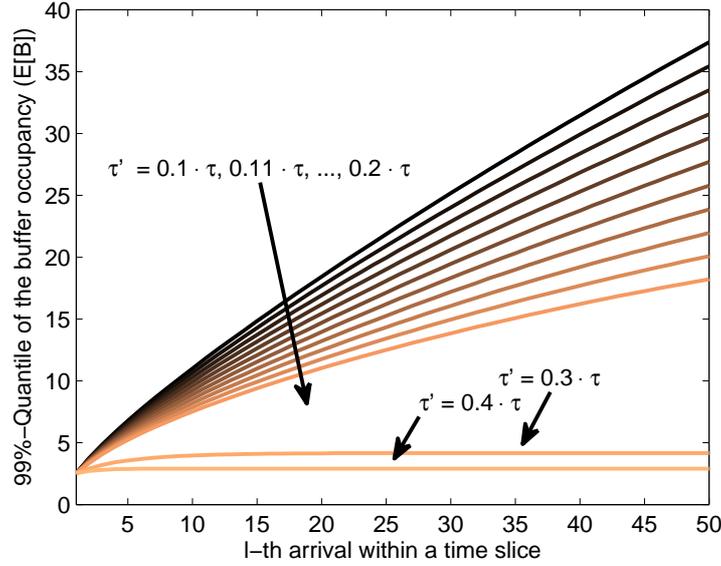


Figure 9: 99%-Quantiles of the buffer occupancy for each arrival and different values of  $\tau'$

to the average processing time  $\tau'^*$ .

For  $\tau' > \tau'^*$  the mean buffer occupancy is  $E[B]$  independent of the transmission phase length. The probe inter-arrival time is large enough to allow a complete processing of the measurement data of one probe before the next data set arrives. At most one data set has to be stored in the queue. Since the reaction time is dominated by the processing time of the last probe data, it increases linearly with the transmission phase.

The influence of the variation of  $\tau'$  on the queue sizes  $o_l$  at each arrival within a time slice is illustrated in Fig. 9. For the case of  $\rho = 0.25$ , the 99%-quantiles as multiples of  $E[B]$  for values of  $\tau' = 0.1, \dots, 0.4 \cdot \tau$  are shown. Thus, the interarrival time between two requests varies between  $\{\frac{0.1}{50} \cdot \tau; \frac{0.4}{50} \cdot \tau\}$ . The x-axis denotes the index of the  $l$ -th arrival within a time slice.

For  $\tau' = 0.1 \cdot \tau$ , the buffer occupancy increases less than linear with the number  $l$  of arrivals within the transmission phase. That is due to the fact that, for  $\tau' < \tau'^*$ , the average inter-arrival time is smaller than the average processing time of a request. Thus, the system fills up and later arrivals tend to see more unfinished work in the system. For higher values of  $\tau'$ , the transmission phase increases, and thus the inter-arrival time between two arrivals, too. This leads to a slower increase of the buffer occupancy since more requests can be served within the transmission phase. While a variation of  $\tau'$  has a significant impact on the buffer occupancy for  $0.1 \cdot \tau < \tau' < 0.2 \cdot \tau$ , this effect is reduced for larger values of  $\tau'$ . Hence, increasing  $\tau' > 0.3$  leads only to marginal improvements for the buffer occupancy. This also justifies that we consider the buffer occupancy of the last arrival since all curves are monotonously increasing with the number of arrivals, denoting the worst case in terms of buffer occupancy.

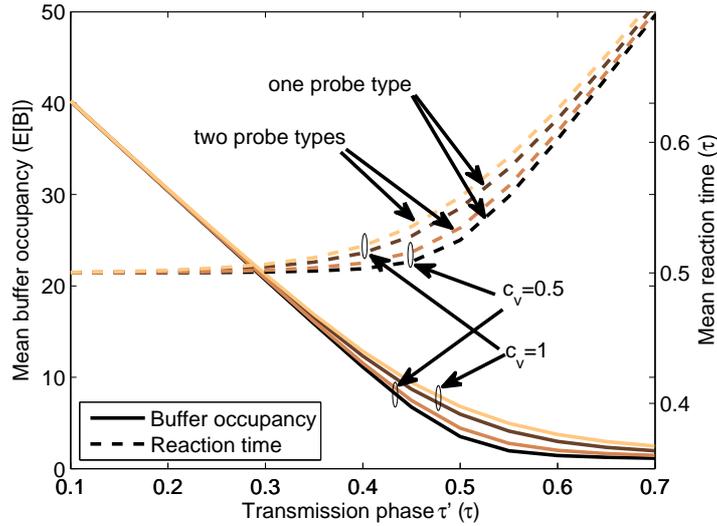


Figure 10: Influence of different coefficients of variation  $c_v$

## 5.2 Influences of Variation on the System

Next we investigate the influence of variation in the work generated by the probes on the buffer occupancy and the reaction time. Again we assume two different negative binomial distributions for the packet size,  $b_i(t)$  and  $b_2(t)$ . First we compare different coefficients of variation  $c_v$  for these distributions, whereas  $E[B_1]$  and  $E[B_2]$  keep constant.

With regard to the probe type, this investigation consists of two scenarios. The first scenario investigates two different probe types. Therefore, it consists of  $s_1 = 25$  probes following a negative binomial distribution with  $E[B_1] = \frac{50}{10000} \cdot \tau$  and  $s_2 = 25$  probes following a negative binomial distribution with  $E[B_2] = \frac{150}{10000} \cdot \tau$ . For the second scenario, we choose one probe type with  $s = 50$  probes following a negative binomial distribution with  $E[B] = \frac{100}{10000} \cdot \tau$ . For these scenarios, both with a system load  $\rho = 0.5$ , we analyzed the behavior of the mean buffer occupancy and the mean reaction time for two coefficients of variation,  $c_v = 0.5$  and  $c_v = 1$ .

The results of this analysis are depicted in Fig. 10 for different lengths of the transmission phase  $\tau' = 0.1, \dots, 0.7 \cdot \tau$ . The solid line again indicates the buffer occupancy while the dashed lines denotes the reaction time.

For the case of  $c_v = 0.5$ , the reaction time, is longer in case of two different probe types as in the case with only one probe type. The average amount of unfinished work per time slice is equal in both cases. However, due to the heterogeneous probe types, the variation is bigger in the case with two different probe types. Thus, the mean buffer occupancy and also the mean reaction time are higher for these scenarios. We also observe that the average time needed for processing the data without idle phase,  $\tau^{I*}$  decreases. That means, that in order to achieve a small average reaction time, the length of the transmission phase has to set to  $\tau' < \tau^{I*}$ . However, for values of  $\tau' < \tau^{I*}$ , the average buffer occupancy increases. This impact becomes more obvious for higher

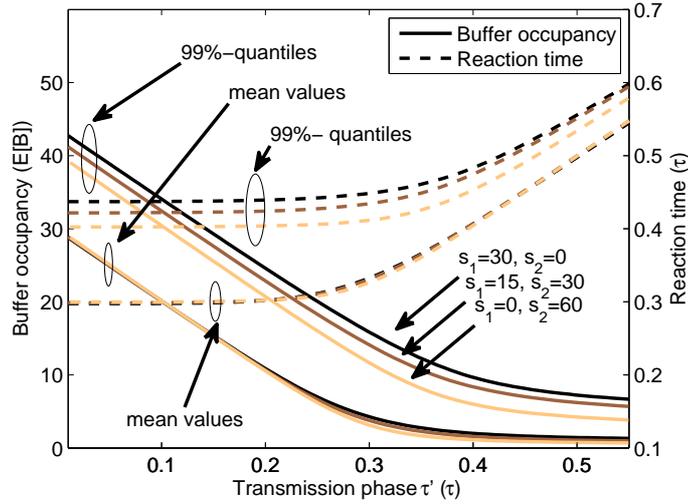


Figure 11: Scenarios for different amounts of probes with constant system load

coefficients of variations. In the case of  $c_v = 1$ , the average reaction time remains constant for values of the transmission phase of  $\tau' \leq 0.3$ . For  $\tau' \geq 0.3$ , the average reaction time increases with respect to the length of the transmission phase  $\tau$ . We can conclude that in case of several probe types increase the variation of the system, and that such an increase has a significant impact on the choice of the transmission phase  $\tau'$ .

The second investigation in this subsection deals with a variation of the amount of probes belonging to the probe types. For each scenario, the overall average packet size  $E[B]$ , and thus the system load  $\rho$  remains constant. We consider two negative binomial packet distributions with  $E[B_1] = \frac{100}{10000} \cdot \tau$  and  $E[B_2] = \frac{50}{10000} \cdot \tau$ . For both distributions we adjust  $c_v = 1$ . The number of  $s_1$  of probes following the distribution  $b_1(t)$  ranges between  $0 \leq s_1 \leq 30$ . The system load is set to  $\rho = \frac{1}{3}$  and the number of probes following  $b_2(t)$  is adjusted in order to reach this load. The influence of three different scenarios on the mean values and the 99%-quantiles of the buffer occupancy and the reaction times for different  $\tau'$  are depicted in Fig. 11. We observe two homogeneous scenarios and a heterogeneous scenario. The first homogeneous scenario consists of  $s_1 = 30$  probes which generate a high amount of work. For the second homogeneous scenario we double the number of probes,  $s_2 = 60$ , and halve the amount of generated work per probe. Hence, the load is constant in both homogeneous scenarios. For the heterogeneous scenario we use  $s_1 = 15$  probes of the first type and  $s_2 = 30$  probes of the second type. That means, that the load keeps constant in all three scenarios.

We observe a clear trend toward shorter reaction times and lower buffer occupancy for a larger number of probes. This effect is negligible for the average values of the buffer occupancy and the reaction time. However, the ratio of probes has a significant impact on the 99%-quantiles. That is due to the varying amount of probes and the constant

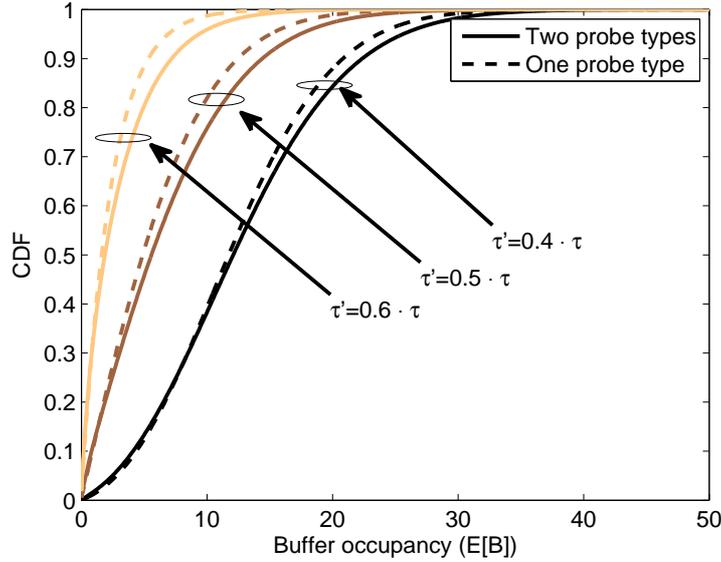


Figure 12: Buffer occupation for the arrival pattern distributed arrival with passive phase for different  $\tau'$ ,  $c_v = 1$

system load. The average packet size  $E[B]$  keeps constant, i.e. the average amount of unfinished work per time slice, and thus the average reaction time and buffer occupancy keep constant. But a higher amount of probes causes a decrease of the variation of the unfinished work per time slice. Thus, the higher quantiles, like the 99%-quantile, decrease. We can conclude that in case of a constant system load  $\rho$  a higher amount of probes generating fewer work leads to faster reaction times and a lower buffer occupancy.

### 5.3 Buffer and Reaction Time Distributions

We now investigate the trade-off between the buffer occupancy and the reaction time for selected values of the transmission phase for a fixed system load of  $\rho = 0.5$  and  $c_v = 1$ . For that we use the same scenarios as presented in the first investigation of Subsection 5.2.

Figure 12 shows the cumulative distribution function of the buffer occupation for  $\tau' = 0.4, 0.5, 0.6 \cdot \tau$ .

We observe that, especially for higher quantiles, the buffer occupancy is higher for the case with two probe types. Although the mean values for the unfinished work are the same in both scenarios, the variation of the unfinished work per time slice is higher for scenario a).

The distribution of the reaction time is shown in Fig. 13. The variation clearly decreases and the average reaction time increases with the transmission phase. The curve with short transmission phases of  $\tau' = 0.4 \cdot \tau$  shows the largest variation which corresponds to the observation that in this case the processing time is dominating the reaction time. In case of  $\tau' = 0.6 \cdot \tau$ , the reaction time has a much smaller variation and depends

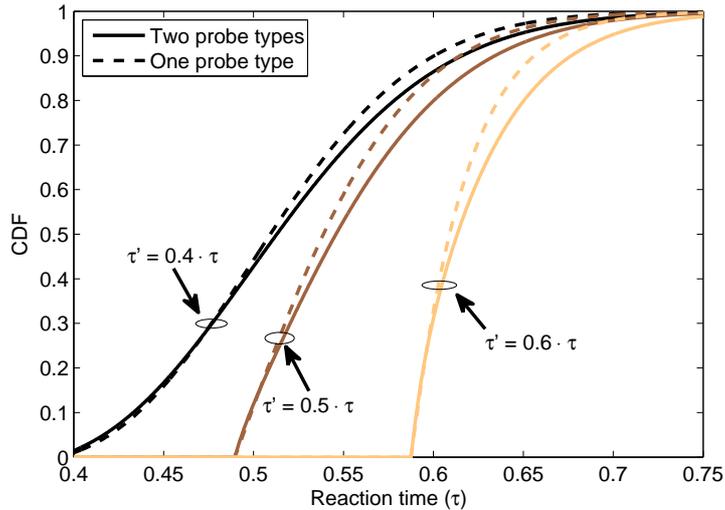


Figure 13: Reaction time for the arrival pattern distributed arrival with passive phase for different  $\tau'$ ,  $c_v = 1$

more on the length of the transmission phase. Note that this behavior is a desirable property for real-time systems. In case of a transmission phase equal to the processing time, the variation is smaller than in case of lower values of  $\tau'$ . The distribution is right-skewed due to the processing of queued data probes from previous arrivals.

The evaluated results can be used for designing the server in an arbitrary multi probe network. For that, we identify the interaction of the reaction time and the buffer occupancy. We assume a buffer size of  $20 \cdot E[B]$  for the system and a desired 90%–quantile for the reaction time of  $0.6 \cdot \tau$ .

From Figure 12 we can conclude, that only for  $\tau' = 0.6 \cdot \tau$  the buffer size requirement is fulfilled. But, from the analysis of reaction times, cf. Figure 13, we observe, that for this value the 90%–quantile of the reaction time is exceeded. Due to the high variation, the requirements can not be guaranteed. In order to fulfill the requirements, the server would have to be dimensioned bigger or the variation of the counter distribution would have to be reduced. Without such an extension,  $\tau' = 0.5 \cdot \tau$  would be most suitable for the system. For a buffer size of  $40 \cdot E[B]$ ,  $\tau' = 0.4 \cdot \tau$  would be possible and speed up the mean reaction time considerable.

## 6 Conclusion & Outlook

The paper developed and investigated a mathematical model for the information sink in large scale multi-probe networks. Currently the transport network constitutes the bottleneck from applications perspective, but for the Future Internet this may change. Especially in the case of multimedia and surveillance applications with a huge number of data sources, bottlenecks will appear at the information sink.

Since specific applications can not yet be foreseen we defined a performance model for

a diversity of probes participating in one system. With the introduced analytical method the reaction times, which denotes the processing duration for the data generated by all probes within the system, can be computed. Furthermore the trade-off between the reaction time and the buffer for different scheduling mechanisms can be investigated. Thus, the presented model can be used by application designers and administrators to design their system for their specific constraints, e.g. for minimizing the reaction time for a given buffer occupancy.

We investigated the system for different system loads, different variations of the counter distributions and different arrival patterns. We further showed how to use the presented results for the design of such a system and explained how to tune the system to fulfill given requirements. An open issue is the investigation of transmission failures and their influence on buffer occupancy and reaction times. Also the model can still be extended by, e.g., including additional random service times denoting the influence of operating systems and other processes.

## Acknowledgments

The authors would like to thank Barbara Staehle for the fruitful discussion throughout the course of this work, as well as Tobias Hossfeld and Frank Lehrieder for their suggestions and improvements.

## References

- [1] GENI Consortium, "GENI - Global Environment for Network Innovations," 2006. Information available at <http://www.geni.net/>.
- [2] N. Gershenfeld, R. Krikorian, and D. Cohen, "The internet of things," *Scientific American*, Oct. 2004.
- [3] UCLA Center for Embedded Networked Sensing, "The Urban Sensing Project," 2008. Information available at <http://urban.cens.ucla.edu/>.
- [4] R. Lasseter and P. Paigi, "Microgrid: a conceptual solution," in *35th IEEE Power Electronics Specialists Conference (PESC'04)*, (Aachen, Germany.), jun 2004.
- [5] K. Eger, C. Gerdes, and S. Öztunali, "Towards p2p technologies for the control of electrical power systems," in *Proc. of the 2008 IEEE Conference on Peer-to-Peer Computing (P2P2008)*, (Aachen, Germany.), sep 2008.
- [6] Z. Yang and K. Nahrstedt, "A Bandwidth Management Framework for Wireless Camera Array," in *Proc. 15th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'05)*, (Stevenson, Wa., USA.), jun 2005.
- [7] U. Helmbrecht, "The role of the Federal Office for Information Security within the German Security policy," 2007. Speech given at Round Table Meeting ÖDigital

Assets in a Cooperative and Mobile WorldÒ at San Francisco, Ca., on Feb. 5th, 2007; Information available at [www.bsi.de/bsi/reden/050207SanFrancisco.pdf](http://www.bsi.de/bsi/reden/050207SanFrancisco.pdf).

- [8] T. Zinner, D. Staehle, P. Tran-Gia, A. Maeder, and K. Tutschku, "Performance evaluation of the information sink in a multi-probe statistical anomaly detection system," in *Proc. of the Australasian Telecommunication Networks and Applications Conference 2008 (ATNAC2008)*, (Adelaide), dec 2008.
- [9] M. Fiedler, K. Tutschku, and P. Carlsson, "Identification of Performance Degradation in IP Networks Using Throughput Statistics," in *Proc. of the 18th International Teletraffic Congress - ITC18*, (Berlin, Germany), sep 2003.
- [10] C. Gessner, "Umts long term evolution (lte) technology introduction," tech. rep., Rohde & Schwarz, Munich, Germany., 2008. Information available at <http://www.rohde-schwarz.com>.
- [11] The 100GET Consortium, "100 GET: 100 Gbit/s Carrier-Grade Ethernet Transport Technologies ," 2008. Information available at <http://www.celtic-initiative.org/Projects/100GET/>.
- [12] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, "Network virtualization: Implementation steps towards the future internet," in *Proc. of "KiVS - Kommunikation in Verteilten Systemen" (KIVS'09)*, (KiVS 2009, Kassel), mar 2009.
- [13] S. Rixner, "Network virtualization: Breaking the performance barrier," *ACM Queue*, Jan./Feb. 2008.
- [14] C. F. Chiasserini, R. Gaeta, M. Garetto, M. Gribaudo, D. Manini, and M. Sereno, "Fluid models for large-scale wireless sensor networks," *Perform. Eval.*, vol. 64, no. 7-8, pp. 715–736, 2007.
- [15] J. Virtamo and J. Roberts, "Evaluating buffer requirements in an ATM multiplexer," in *IEEE Globecom 89 Proceedings*, (Dallas, USA.), Nov 1989.
- [16] K. L., *Queueing Systems, Volume I: Theory*. Wiley Interscience, New York, 1975.
- [17] P. Tran-Gia, "Discrete-time analysis technique and application to usage parameter control modelling in ATM systems," in *Proceedings of the 8th Australian Teletraffic Research Seminar*, (Melbourne, Australia), December 1993.
- [18] P. Tran-Gia, *Einführung in die Leistungsbewertung und Verkehrstheorie*. München, Deutschland: Oldenbourg, July 2006.