[PSSt04] *Krishna Pandit, Jens Schmitt, Ralf Steinmetz;* **Network Calculus Meets Queueing Theory - A Simulation Based Approach to Bounded Queues;** 2004 Twelfth IEEE International Workshop on Quality of Service (IWQoS'04), Montreal, Canada, June 2004, S.

Network Calculus meets Queueing Theory - A Simulation Based Approach to Bounded Queues

Krishna Pandit*, Jens Schmitt† and Ralf Steinmetz*

*Department of Electrical Engineering and Information Technology
Technische Universität Darmstadt, Germany
Email: {Krishna.Pandit, Ralf.Steinmetz}@kom.tu-darmstadt.de

†Department of Computer Science
University of Kaiserslautern, Germany
Email: jschmitt@informatik.uni-kl.de

Abstract— **Quality of Service (QoS) is an area with high academic curiosity. Our long-term goal is to develop a unified mathematical model. This paper is a first step towards this ambitious goal. The most widespread models for network QoS are Network Calculus and Queueing Theory. While the strength of Queueing Theory is its proven applicability to a wide area of problems, Network Calculus can offer performance guarantees. We analyse by simulation the benefit of bringing the two of them together, i.e., bounding the stochastic processes of a queue with methods from Network Calculus. A basic result from Network Calculus is that enforcing traffic shaping and service curves bounds the buffer. This leads to denying buffer states in queues with infinite buffer. Specifically, we analyse what happens with the probability mass of such buffer states. Finally, we discuss how our results can be used for dimensioning buffers for multiplexed traffic.**

I. Introduction

A. Motivation

Despite recent doubts and frustrations, Quality of Service (QoS) in the Internet remains a much debated research issue. QoS research can be divided into two classes: administrative vs. technical issues. The former includes aspects such as pricing, accounting, security and the interconnection between service providers. Technical issues are the actual manipulation of the data packets, such as traffic regulation, scheduling and admission control algorithms. In the context of technical issues, most research to date has gone into developing and optimising new architectures and algorithms. An open research issue, which is relatively underexposed, is a unified theoretical model for QoS. The two front runner models for QoS in packet networks undoubtedly are Queueing Theory and Network Calculus. We assume that in future there will be some kind of traffic shaping and policing in the Internet since this is essential to offer any kind of service guarantees, be they deterministic or statistical. We believe the path to a unified model for QoS consists of bringing Network Calculus and Queueing Theory together. In this paper, we take the first step by conducting a simulative approach to analyse the impact of Network Calculus bounds on Queueing Theory results. This could be viewed as a Network Calculus-assisted Queueing Theory.

The remainder of this paper is organised as follows. In the following subsections we review some basic results from Network Calculus and discuss the related work. In section II we introduce the system model. We then conduct the simulations and discuss them in section III. In section IV the simulation results are given. Finally, we conclude and give an outlook.

B. Background

While we restrict the discussion of Queueing Theory to a few general remarks, we recapitulate the definitions and theorems of Network Calculus used in this paper in this subsection.

In Queueing Theory, generally, the average quantities in an equilibrium state are considered. However, obtaining a rich set of tractable results comes at the cost of having to restrict to Markovian (memoryless) traffic. Beginning with [9], it has been shown several times that this is not necessarily a realistic assumption for Internet traffic. Another drawback which could be mentioned is that there are few results on the transient analysis of queueing systems. We assume that the reader is familiar with Queueing

[1]This work was partly funded by the Deutsches Forschungsnetz Verein (German Research Network) as part of the LETSQoS project.

[2]Accepted for publication at the IEEE International Workshop on Quality of Service (IWQoS), June, 2004.

Theory and refer to [7] as an excellent book.

Network Calculus [6] is a theory for deterministic queueing systems. The underlying idea is that service guarantees can be achieved by regulating the traffic and deterministic scheduling. Analogous to conventional system theory, a system consists of an input, a transfer function and an output. The input, mostly referred to as *arrival curve*, is an abstraction of the traffic regulation, and the transfer function, mostly referred to as *service curve*, is an abstraction of the scheduling. The difference to conventional system theory is that the dioid $\{\Re \cup \infty, \min, +\}$ is used, i.e., that addition and multiplication are replaced by minimum and addition, respectively. This is often referred to as *Min-plus Algebra*. The reason to switch to Min-plus Algebra is that this way linearity is preserved. In the following, we recapitulate the results from Network Calculus which are relevant for this paper. They can all be found in the excellent text of Le Boudec and Thiran [3]. As in conventional system theory, a key operation in Network Calculus is the min-plus convolution. Note that the infimum (inf) is similar to the minimum (min), with the sole difference that it does not have to be in the set. The same applies for the supremum (sup) and maximum (max). The min-plus convolution of f and g is the function

$$(f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\} \qquad (1)$$

The traffic bound is given by an *arrival curve*, which denotes the largest amount of traffic allowed to be sent in a given time interval.

Definition 1 (Arrival Curve): Given a wide-sense increasing function α defined for $t \geq 0$, we say that a flow R is constrained by α iff for all $s \leq t$

$$R(t) - R(s) \leq \alpha(t-s)$$

We say that R has α as an arrival curve, or also that R is α-smooth.

The arrival curve can be viewed as an abstraction of the regulation algorithm. The most prominent example for a traffic regulation algorithm is the Leaky Bucket [14], which is often also referred to as Token Bucket. Its arrival curve is given by the following equation.

$$\alpha(t) = b + rt \text{ for } t > 0 \qquad (2)$$

Therefore, no more than b data units can be sent at once and the long-term rate is r.

A *greedy shaper* with the shaping curve σ optimally delays packets, so that the output has σ as an arrival curve, and sends all bits as soon as possible.

Theorem 1 (Greedy Shaper): Consider a greedy shaper with shaping curve σ, which is sub-additive and $\sigma(0) = 0$. Assume that the shaper buffer is empty at time 0, and that it is large enough so that there is no data loss. For an input flow R, the output R^o is given by

$$R^o = R \otimes \sigma \qquad (3)$$

We omit the proof as it can be found in [3].

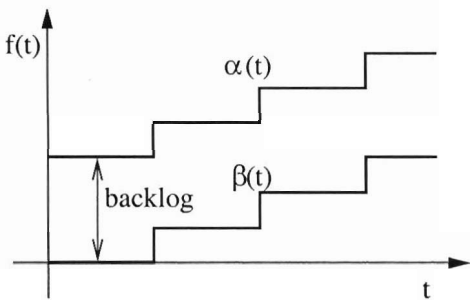

Fig. 1. Backlog

The service curve is an abstraction of the scheduling.

Definition 2 (Service Curve): Consider a system S and a flow through S with input and output functions R and R^o, respectively. We say that S offers to the flow a service curve β if and only if $\beta \in F$ and $R^o \geq R \otimes \beta$.

Due to its application in the Integrated Services context, a prominent service curve is the rate-latency function.

Definition 3 (Rate-latency functions $\beta_{R,T}$):

$$\beta_{R,T} = R[t-T]^+ = \begin{cases} R(t-T) & \text{if } t > T \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

for some $R \geq 0$ (the 'rate') and $T \geq 0$ (the 'delay').

We next introduce the *Backlog Bound*, which is one of the three basic bounds of Network Calculus.

Theorem 2 (Backlog Bound): Assume a flow, constrained by arrival curve α, traverses a system that offers a service curve β. The backlog $R(t) - R^o(t)$ for all t satisfies:

$$R(t) - R^o(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\} \qquad (5)$$

We omit the proof as it can also be found in [3].

On a final note, a drawback of Network Calculus is that it deals with the worst-case behavior of traffic flows, which leads to severe under-utilisation in realistic environments.

C. Related Work

There are several approaches to extend Network Calculus into a stochastic setting. Bounds for the multiplexing of flows are obtained by utilizing methods

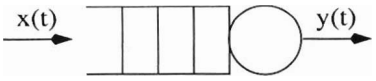

Fig. 2. Queue

Fig. 3. Shaped queue

such as the law of large numbers and the Chernoff Bound. Excellent overviews for this topic are [8] and [5].

A framework for statistically aggregating flows is given in [11]. However, there the focus is on reducing state complexity in the network.

Liebeherr et al. [10] introduced the concept of Statistical Network Calculus. In the first incarnation this is based on the assumption that an arrival curve does not deterministically bound the incoming traffic but bounds it only with a certain probability. Similarly, a statistical service curve [4] is a service curve that only offers the service with a certain probability. Statistical Network Calculus is the most closely related work to our work. It can be seen as an approach to the same goal from a different angle. To recall, the goal is to obtain a model more strict than average behavior but looser than worst-case. Statistical Network Calculus has Network Calculus as a starting point and enhances it with probabilistic methods. We start with Queueing Theory, i.e. a purely probabilistic model, and enhance it with methods from Network Calculus.

Schmitt [12] compares Network Calculus and Queueing Theory results for priority queueing. Furthermore, based on the Network Calculus results, performance bounds can be obtained by enforcing admission control in each priority class [13].

In [1] a shaper is derived that ensures that the traffic has better stochastic properties than a reference process, for which they use the Poisson process. In contrast, our goal is not to derive a shaper, but to assume a Network Calculus based shaper being present and analysing its effect.

To our knowledge there exists no work which is closely related enough to allow a comparison of our numbers. The closest to an analytical solution of this problem can be found in the book by Baccelli et al. [2], and there especially the chapter on Stochastic Event Graphs.

II. System Model

In this section we introduce the system model. Figure 2 shows a traditional queue, consisting of a buffer and server. The input process is given by $x(t)$ and the output process by $y(t)$. For the $M/M/1$ case the input and output processes are both Poisson processes. We now introduce the basic Network Calculus elements to the queue. These are a shaper to manipulate the input process and a service curve enforcer to manipulate the service times. This is depicted in Figure 3. According to Equation 3, $x'(t)$ is obtained by the min-plus convoluiton of $x'(t)$ and the shaping curve, for which we use the Leaky Bucket from Equation 2. In other words, the shaper works as follows. When a packet arrives at the shaper, the shaper checks whether there are enough tokens in the shaper - without loss of generality we assume all packets to be of size 1 throughout this paper - to admit the packet. If that is the case, then the packet traverses the shaper infinitely fast and arrives at the queue. If there are not enough tokens to admit the packet, the packet is held in the shaper until enough tokens have been collected. Therefore, the shaper theoretically has an infinitely large buffer. Since packets are delayed, the shaper might decrease the rate of the process. We define a new arrival rate

$$\lambda' = \frac{\text{\# of packets}}{\text{duration of observation}}$$

Note that in our model the shaper is only a conceptual model rather than an actual device holding packets. We assume that a higher layer such as the application layer ensures that all traffic is conform. When a packet arrives at the queue it checks whether the server is available. If this is the case, it receives service immediately, otherwise it waits in a queue until the server becomes available. Upon arrival of a packet, the server assigns it an exponentially distributed service time. The service curve enforcer then checks whether the service time is less or equal to the maximum service time allowed by the service curve. If this is the case, the service time remains untouched, else the service time is set to the maximum allowed service time. The service curve enforcer therefore increases the server rate. Accordingly, we define a new server rate

$$\mu' = \frac{1}{\text{mean of the actual server rates}}$$

Note that while the shaper can only delay packets, the service curve enforcer releases packets ahead of schedule. Therefore, its placement behind the server seems counter-intuitive as the transfer function mapping $y'(t)$ to $y(t)$ is non-causal. As shaper we use the Token Bucket. Our service curve of choice is the rate-latency curve from Definition 3, which we refer to as RLC hereafter.

We call packets, which are delayed by the shaper, shaper manipulated packets. Accordingly, we call

packets, which are served earlier due to the service curve, server manipulated packets.

Therefore, the parameters of our system model are $(\lambda, \mu, b, r, L, R)$, which denote the arrival rate, service rate, leaky bucket depth, leaky bucket rate, latency of the RLC and rate of the RLC, respectively.

The analysis of the shaper manipulation itself is tedious. Many favourable properties, such as memorylessness and the stationarity, are lost by shaping the Poisson process. It is obvious that the server manipulation is even less tractable, as it depends on the state of the queue. Therefore, a mathematical analysis of this problem is beyond the scope of this paper.

III. Simulations

Qualitatively, we expect the following behaviour in the simulations of the bounded queue. Trivially, the probability of the states higher than the Backlog Bound from Theorem 2 will be 0. The probability of state 0 will remain unchanged in the bounded queue. The reason for this is that the shaper and the service curve enforcer are both inactive when the system is empty. There will be a strong increase in probability mass at the state 1, due to the shaper. The shaper causes the inter-arrival times of packets at the queue to be more equally distributed than in a pure exponential distribution. Note that asymptotically, i.e., when $r \ll \lambda$, all inter-arrival times are $\frac{1}{r}$ after the initial tokens in the bucket have emptied. Both, the traffic shaper as well as the service curve enforcer, cause the probability mass to shift towards the lower states. Therefore, the higher states of the bounded queue will be less probable than the same states of the M/M/1 queue.

As parameters of the simulation we use $(\lambda, \mu, b, r, L, R) = (2, 3, 6, 2, 1, 2)$ Using Theorem 2 we obtain that the maximum buffer state is 7. There are 5000 arrivals per run and the simulation is repeated 30 times. The values are depicted in Figure 4.

The average number of input and shaper manipulated packets are 4933 and 276.2, respectively. As a reference, the state probabilities of the corresponding M/M/1 Queue, i.e., with $\rho' = \frac{\lambda'}{\mu'} = \frac{1.85}{3.18} = 0.58$, are given. Since the difference to the M/M/1/N queue is marginal, we can neglect it. What is striking here is that the probabilities of the high states of the Bounded Queue are lower than of those states in the M/M/1 case. The probability mass of the higher states is neither distributed evenly among the allowed states, nor is it collected in the last allowed state. This result confirms our assumptions, that by putting structure in form of input shaping and service curve enforcement, the behavior of the

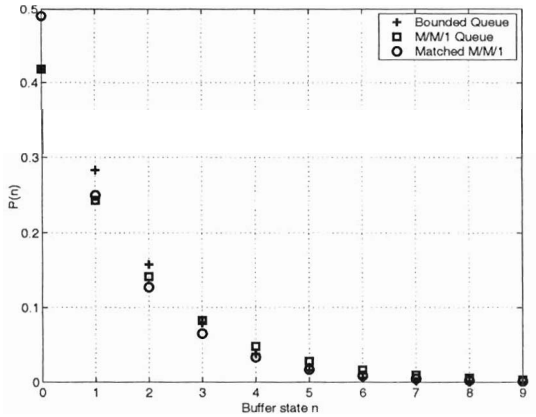

Fig. 4. Bounded Queue vs. M/M/1

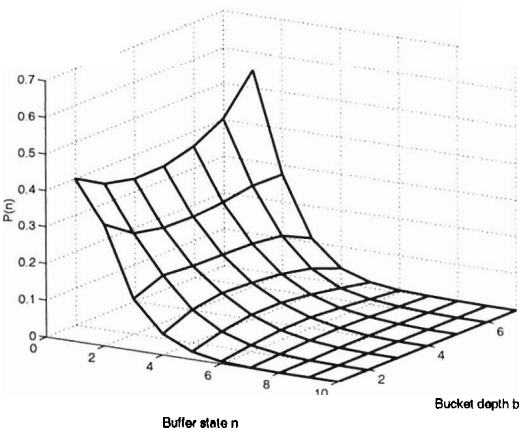

Fig. 5. Bounded buffer occupancy distribution

queue in the relevant buffer states becomes better. Being interested in the state 5, we find that the queue has a better behavior than a M/M/1 queue with $\rho_{match} = 0.51$. This gives us an adjustment factor

$$a = \frac{\rho_{match}}{\rho'} = 0.88$$

We now analyse several parameter sets in order to get an insight on the adjustment factor. We hold the parameters $\lambda = \frac{2}{3}$ and $\mu = 1$ and set the Token Bucket rate equal to the RLC rate $r = R = 1$. In order to compare the buffer occupancy distributions in a fair manner, we ensure that the Backlog Bound is constant at 7. We therefore set $b = 1, 2, \ldots, 7$ and accordingly $L = 7, 6, \ldots, 1$. The buffer occupancy density functions are shown in Figure 5. In Figure 6 some interesting values are shown. These are ρ', λ', μ' and ρ_{match}. As reference, λ and μ are also shown. It can be seen that when the bucket depth is low, and consequently the latency is high, μ' is close to μ. The same applies to λ' vice versa. In Figure 7 the adjustment factor is plotted. It can be seen that it is lowest for the endpoints. This implies that

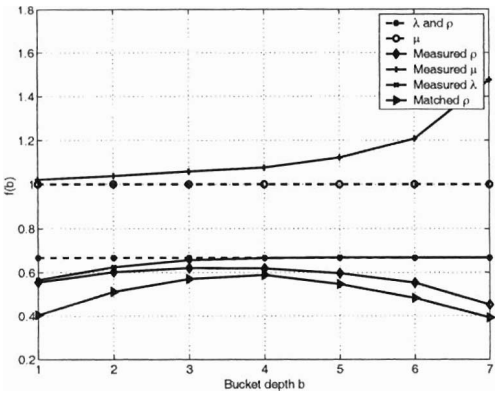

Fig. 6. Relevant ρ, λ and μ

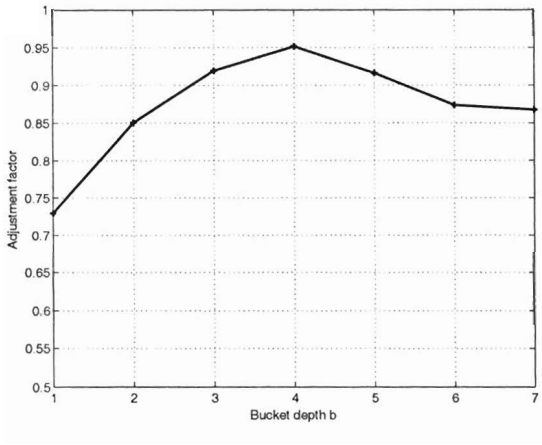

Fig. 7. Adjustment factors

tight shaping or tight service curve enforcement have a stronger influence on the adjustment factor than some shaping and some service curve enforcement combined. The influence of tight shaping is stronger than that of tight service curve enforcement.

IV. Conclusion

In this paper we analyse the impact of traffic shaping and service curve enforcement on a M/M/1 queue. We show how the probability mass of the higher buffer states of the M/M/1 queue distributes over the lower buffer states. We show that the probability mass of the bounded queue strongly shifts towards the lower buffer states. The higher states of the bounded queue are less probable than the same states in the M/M/1 queue. This is a key contribution of this paper as it can be utilised when dimensioning aggregate buffers for multiplexed flows. With the knowledge of the queue being bounded, a lower utilisation than the reference M/M/1 queue can be used. Unfortunately, it was not possible to quantify this effect, due to the complexity of the system. This will be subject of future work. Another obvious issue is an analytical solution to this problem. Further, as the parameter space is large, arbitrarily many simulations can be run. An especially interesting issue would be the impact of different traffic shapers, such as a TSpec, and different service curves. Finally, the setting can be expanded to Queueing Networks, starting with a simple concatenation of queues. There the question arises how simulating the network with a concatenation of nodes, each enforcing a service curve, compares to assuming one node which offers the network service curve. A long shot is considering complex networks with feedback to model flow control.

Acknowledgements

The authors thank Andreas Faatz and Matthias Priebe for their contributions to this paper.

References

[1] D. Abendroth and U. Killat. Intelligent Shaping: Well Shaped Throughout the Network? In *Proceedings of IEEE INFOCOM*, 2002.

[2] F. Baccelli, G. Cohen, G.-J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. John Wiley and Sons, 1992.

[3] J.-Y. L. Boudec and P. Thiran. *Network Calculus*. Number 2050 in Lecture Notes in Computer Science. Springer-Verlag, 2001.

[4] A. Burchard, J. Liebeherr, and S. Patek. A Calculus for End-to-end Statistical Service Guarantees. Technical Report CS-2001-19, University of Virginia, Department of Computer Science, 2002.

[5] C.-S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[6] R. Cruz. A Calculus for Network Delay. I. Network Elements in Isolation. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[7] L. Kleinrock. *Queuing Systems, Volume 1: Theory*. John Wiley and Sons, 1975.

[8] E. Knightly and N. Shroff. Admission Control for Statistical QoS: Theory and Practice. *IEEE Network Magazine*, 13:20–29, March/April 1999.

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

[10] J. Liebeherr, S. D. Patek, and E. Yilmaz. Tradeoffs in Designing Networks with End-to-End Statistical QoS Guarantees. In *Proceedings of the IEEE/IFIP Eighth International Workshop on Quality of Service (IWQoS '2000)*, pages 221–230, 2000.

[11] K. Pandit, J. Schmitt, and R. Steinmetz. Aggregation of Heterogeneous Real-Time Flows with Statistical Guarantees. In *Proceedings of 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'02), San Diego, USA*, pages 57–64. SCS, July 2002. ISBN 1-56555-252-0.

[12] J. Schmitt. On Average and Worst Case Behaviour in Non-Preemptive Priority Queueing. In *Proceedings of the 2003 Internation Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pages 197–204, 2003.

[13] J. Schmitt, P. Hurley, M. Hollick, and R. Steinmetz. Per-flow Guarantees under Class-Based Priority Queueing. In *Proceedings of IEEE Global Telecommunications Conference 2003 (GLOBECOM 2003), San Francisco, CA, USA*, Dec. 2003.

[14] J. Turner. New Directions in Communications (or Which Way to the Information Age?). *IEEE Communications Magazine*, 24(10):8–15, Oct 1986.