Stochastic Network Calculus Latency Bounds

Curt Polack, Max Helm*, Dominik Scholz*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: curt.polack@tum.de, helm@net.in.tum.de, scholz@net.in.tum.de

Abstract—

The question of how to calculate latency and buffer bounds in complex networks is becoming ever more relevant in today's interconnected world. A number of approaches to this problem have been developed; this paper provides an introduction to the concepts as well as the application of stochastic network calculus and shortly introduces two alternative methods for calculating relevant bounds in networks: deterministic network calculus and classic queuing theory. This paper also provides an overview of various open-source tools that use these different approaches and, using two different reference topologies, compares them regarding important factors such as the tightness of bounds. While the tested tools perform similarly in uncongested networks, specific tools, such as DISCO SNC, can provide tighter bounds and greater functionality. The results of one tool, DISCO DNCv2, could not be reliably compared with those of the others.

Index Terms—**stochastic network calculus, queuing theory, network performance evaluation, scheduling**

1. Introduction

The increasing dependence on fast, reliable networks in academia and elsewhere necessitated the formalization of those networks. Quality of Service (QoS) guarantees regarding delay, throughput, and packet loss are especially important when measuring the performance of networks and ensuring their QoS [1] . The required guarantees vary depending on the needs of end-users and the contracts they have with their respective network provider. A user who, for example, is interested in real-time communication with other users would be less interested in the lower bounds on throughput and more interested in the upper bounds on delay; on the other hand, a user who must send large files would be more concerned with the lower bounds on throughput.

Network calculus was first introduced in 1991 by Cruz in two related papers, [2] and [3], as a new way to determine the relevant bounds inside a communication network. Since this initial publication, network calculus has developed in two branches: stochastic and deterministic network calculus. In short, deterministic network calculus provides the worst-case bounds for a given network while stochastic network calculus provides bounds based on statistical distributions and a certain level of acceptable exceedance of required bounds [4]. Deterministic network calculus is often used to model networks in which there is no tolerance for packet loss or long delays (e.g. critical infrastructure). Stochastic network calculus can be employed to model networks which can tolerate a certain amount of loss or delay in order to more closely mirror the real-life requirements of many networks and users. It can, therefore, be used to more accurately and tightly find bounds / guarantees in networks that are inherently statistical in nature [4].

2. Background

This section provides an overview of the fundamentals of (stochastic) network calculus and its similarities and differences with other theories.

2.1. Network Calculus

In order to be effectively used for network analysis (i.e. deriving relevant bounds), a theory must be characterized by the following five properties [4]:

1) Service Guarantees - Stochastic service guarantees (e.g. backlog, delay) can be derived for single nodes using a specific traffic model and a specific server model.
2) Output Characterization - The output of a server can be modeled using the same traffic model as the input.
3) Concatenation - The concatenation (i.e. convolution) of multiple servers can be modeled using the same server model.
4) Leftover Service - The service available to a traffic flow can be modeled using the same server model if multiple flows are simultaneously using the service.
5) Superposition - The superposition of multiple traffic flows can be modeled using the same traffic model.

These properties, especially the third and fourth properties, allow for the concatenation of multiple service and arrival flows; this, in turn, reduces the necessary calculations and can significantly improve the results when compared to node-to-node analysis [4].

Network calculus characterizes networks using two curves: service (server) and arrival (traffic). The arrival curve describes the traffic sent using its upper bound while the service curve defines a lower bound that a server provides [4]. Often an "envelope process" is used to describe these curves; the envelope process simply

doi: 10.2313/NET-2020-11-1_16

refers to a function which deterministically bounds the process but is not necessarily tight. One of the main advantages of network calculus is that service curves can be concatenated (i.e. convoluted) using min-plus algebra and thus more effectively analyzed.

2.2. Mathematical Basics & Notation

This section provides an overview of the notation used and, partially, the fundamental mathematical concepts as described and used in [4] and [5].

$\mathbb{F}$ is used to denote the set of non-negative wide-sense increasing functions in $\{a() : \forall 0 \leq x \leq y, 0 \leq a(x) \leq a(y)\}$ and for which it holds $\forall x < 0 : a(x) = 0$. This set of functions is used to characterize arrival and service curves as a function of time t.

Min-plus algebra is used to perform operations on flows. An important algebraic structure when using min-plus algebra is

$$(\mathbb{F} \cup \{+\infty\}, +, \wedge)$$

which is a commutative diode with the zero element $+\infty$ and the identity element 0 for all $x \geq 0$ and $+\infty$ otherwise [6]. The min-plus convolution and deconvolution of two functions, a and b, are respectively defined as follows:

$$(a \otimes b)(x) = \inf_{0 \leq x \leq y} [a(x + y) - b(y)]$$

$$(a \oslash b)(x) = \sup_{y \geq 0} [a(x + y) - b(y)]$$

$A(t)$ refers to the (cumulative) arrival process and $A*(t)$ to the (cumulative) departing traffic process of a (lossless) server. The backlog $B(t)$ is then defined as $A(t) - A*(t)$. A^i_h and $A{*}^i_h$ refer respectively to the arrival and departure models of flow i in network element h.

2.3. Traffic Models

The traffic model originally introduced in [2] is the (σ, ρ) traffic characterization and is deterministic. σ refers to the burstiness and ρ to the rate of the traffic flow. A popular implementation of this traffic flow is referred to as a token bucket; a traffic flow / arrival curve A is bounded by the (σ, ρ) model if the following condition holds for all $0 \leq s \leq t$ [4]:

$$A(s, t) \leq \rho \cdot (t - s) + \sigma$$

In stochastic network calculus, this model is expanded and depicted using the $(\sigma(\theta), \rho(\theta))$ traffic characterization [4]. Using the moment generating function (MGF) of the arrival curve, a bound can be derived for this traffic characterization [4]; an arrival curve A is bounded by the $(\sigma(\theta), \rho(\theta))$ model for some θ if the following condition holds for all $0 \leq s \leq t$:

$$\frac{1}{\theta} logE[e^{\theta A(s,s+t)}] \leq \rho(\theta) \cdot (t) + \sigma(\theta)$$

Stochastic network calculus can then be used to find and optimize θ by defining, for example, the maximum allowed backlog and the probability with which this bound must be respected [5]. There are a number of further variations of the stochastic arrival curve introduced in [4] including the traffic-amount-centric (t.a.c.) arrival curve, the virtual-backlog-centric (v.b.c.) arrival curve, and the maximum-backlog-centric (m.b.c.) arrival curve.

2.4. Similarities and Differences with Queuing Theory

Queuing theory was first introduced in 1909 by A.K. Erlang and is a branch of mathematics that focuses on the modelling of the act of waiting in line [7]; queuing theory was thus not developed with modern communication / packet networks in mind. One of the most common proposals to model arrival and service curves is the M/M/1 model using a Poisson distribution. Although many networks can be modeled using queuing theory, multiple of the properties, especially 3 and 4, described in Section 2.1 cannot, in general, be concluded for queuing theory [4]. In addition, queuing theory focuses on the calculation of the average case and not the worst case, as in network calculus [7].

3. Application and Comparison

This section provides an overview of a diverse set of tools as well as their defining characteristics and defines two reference topologies. The aforementioned tools are then used to evaluate both reference topologies; the results of all evaluations are subsequently analyzed and compared with one another.

3.1. Tools

The tools which are introduced in this section are all open-source and were developed primarily for use in research. Table 1 provides an overview of the supported network topologies (Tandem, Tree, and Feed-forward) as well as important network calculus operations (Convolution and Deconvolution).

TABLE 1: Supported Topologies and Operations

	Tandem	**Tree**	**Feed-forward**	$\otimes$	$\oslash$
DISCO SNC	✓	✓	✓	✓	✗
DISCO DNCv2	✓	✓	✓	✓	✓
SNC MGF Toolbox	✓	✓	✗	✓	✓
OMNeT++	✓	✓	✗	✗	✗

3.1.1. The DISCO Stochastic Network Calculator. The DISCO Stochastic Network Calculator (DISCO SNC) is an open-source tool developed by researchers at the Distributed Computer Systems (DISCO) chair of the University of Kaiserslautern [8]. It is a modular program developed primarily in Java for the application of stochastic network calculus and also provides a graphical user interface (GUI) in order to "make the SNC accessible even for SNC-inexperienced users" [8]. It supports a number of traffic characterizations including exponential, exponentially bounded burstiness, and token bucket.

3.1.2. Stochastic Network Calculus Moment Generating Function Toolbox. The Stochastic Network Calculus Moment Generating Function Toolbox (SNC MGF Toolbox) is an open-source tool developed by Paul Nikolaus, a researcher at the DISCO chair of the University of Kaiserslautern [9]. It is a library developed in Python for the application of stochastic network calculus and does

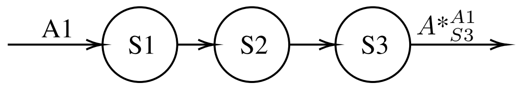

Figure 1: Topology 1 - Tandem

not provide a GUI. It supports a number of traffic characterizations, including Markov modulated on-off traffic, exponentially bounded burstiness, and token bucket.

3.1.3. The DISCO Deterministic Network Calculator v2. The DISCO Deterministic Network Calculator v2 (Disco DNCv2) is the second-generation open-source tool developed by researchers at the DISCO chair of the University of Kaiserslautern for analyzing networks using deterministic network calculus [10].

3.1.4. Objective Modular Network Testbed in C++. Objective Modular Network Testbed in C++ (OMNeT++) is a discrete event simulator developed primarily in C++ and used for, among other things, modeling communication networks [11]. OMNeT++ was first introduced in 1997 and has since been expanded by multiple libraries, including one for the simulation of queuing networks, and a comprehensive GUI.

3.2. Reference Topologies

In order to compare the tightness of the bounds calculated by the aforementioned tools, this section defines two relatively simple network topologies which can be modeled in every tool. Figure 1 models a tandem (chain) network with three service curves; the arrival curve enters the first node and is processed by all three service curves. Figure 2 models a (fat) tree network with two arrival curves entering the two lowest nodes respectively and both departing curves being sent to the root node / service curve. S1, S2, and S3 provide a constant service rate of 5, 4.9, and 4.5 respectively in both topologies. The bounds on the departure curve(s), $A*^{A1}_{S3}$ for the tandem topology as well as $A*^{A1}_{S1}$ and $A*^{A2}_{S1}$ for the tree topology, are of interest for the analysis. The arrival curve(s) A1 and A2 are characterized by an exponential distribution with $exp(\lambda)$; multiple values for λ are tested. For both stochastic network calculus tools, the tolerance for the delay as well as the backlog bound was set to .05. The graphed results for the tree topology are cumulative results of both $A*^{A1}_{S1}$ and $A*^{A2}_{S1}$.

The arrival curve(s), which can be represented by the exponential distribution with the parameter λ, are bounded by the $\sigma(\theta), \rho(\theta))$ traffic characterization for all $\theta < \lambda$ with $\sigma(\theta) = 0$ and $\rho(\theta) = \frac{1}{\theta}(\frac{\lambda}{\lambda-\theta})$ [12]. This can be used to provide a deterministic bound on the stochastic distributions for DISCO DNCv2.

3.3. Configuration of Tools

This section documents the methods used to configure each tool and derive relevant bounds using the tandem topology and backlog bound as an example.

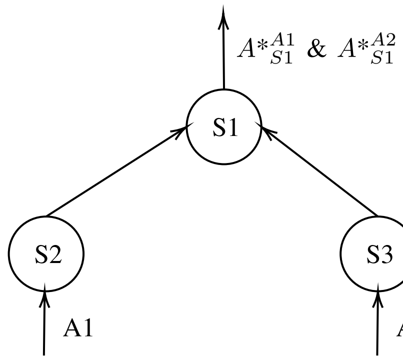

Figure 2: Topology 2 - Fat Tree

3.3.1. DISCO SNC. The following configuration can be loaded directly into the DISCO SNC GUI for $\lambda = .3$.

```
# Configuration of Network
# Interface configuration. Unit: Mbps
I v1, FIFO, CR, 5
I v2, FIFO, CR, 4.9
I v3, FIFO, CR, 4.5

EOI
# Traffic configuration. Unit Mbps or Mb
# One flow with the route v1->v2->v3 with priorities and characterization
F F1, 3, v1:1, v2:1, v3:1, EXPONENTIAL, .3
EOF
```

3.3.2. SNC MGF Toolbox. This Python file for $\lambda = .3$ can be run in the root folder of the project once all dependencies have been added.

```
if __name__ == '__main__':
    print("Tandem␣Performance␣Bounds:\n")

    DELAY_PROB_BOUND = PerformParameter(perform_metric=PerformEnum.DELAY_PROB,
    value=.05)

    Server1 = ConstantRateServer(rate=5.0)
    Server2 = ConstantRateServer(rate=4.9)
    Server3 = ConstantRateServer(rate=4.5)

    ConvolvedServer = Convolve(Convolve(Server1, Server2), Server3)

    TandemTopology = SingleServerMitPerform(arr_list=[DM1(lamb=.3)],
                                            server=ConvolvedServer,
                                            perform_param=DELAY_PROB_BOUND)

    #Grid search for param between 0.1 and 5.0 with granularity 0.1
    print(Optimize(TandemTopology, number_param=1,
                   print_x=True).grid_search(bound_list=[(0.1, 5.0)],
                   delta=0.1))
```

It is important to note that SNC MGF Toolbox does not support FIFO multiplexing / scheduling and instead always uses arbitrary scheduling. Arbitrary scheduling provides a worst-case bound on any scheduler, including FIFO.

3.3.3. DISCO DNCv2. This Java file for can be run for $\lambda = .3$ and $\theta = .1$ in the root folder of the project once all dependencies have been added.

```
public void run() throws Exception {
    ServiceCurve service_curve_1 = Curve.getFactory()
    .createRateLatency(5.0, 0);
    ServiceCurve service_curve_2 = Curve.getFactory()
    .createRateLatency(4.9, 0);
    ServiceCurve service_curve_3 = Curve.getFactory()
    .createRateLatency(4.5, 0);

    ServerGraph sg = new ServerGraph();

    Server s0 = sg.addServer(service_curve_1, Multiplexing.FIFO);
    Server s1 = sg.addServer(service_curve_2, Multiplexing.FIFO);
    Server s2 = sg.addServer(service_curve_3, Multiplexing.FIFO);

    sg.addTurn(s0, s1);
    sg.addTurn(s1, s2);

    ArrivalCurve arrival_curve = Curve.getFactory().createTokenBucket(15, 0);
    sg.addFlow("f0", arrival_curve, s0, s2);
    CompFFApresets compffa_analyses = new CompFFApresets( sg );
    TotalFlowAnalysis tfa = compffa_analyses.tf_analysis;

    tfa.performAnalysis(sg.getFlow(0));
    System.out.println("delay␣bound:␣" + tfa.getDelayBound());
    System.out.println("backlog␣bound:␣" + tfa.getBacklogBound());
}
```


doi: 10.2313/NET-2020-11-1_16

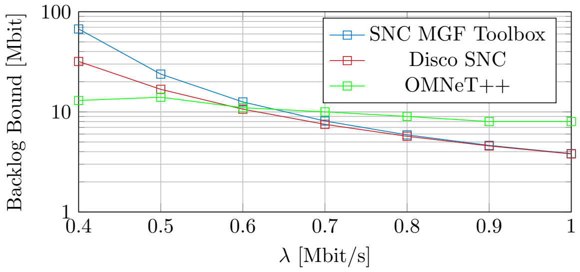

Figure 3: Backlog Bound - Tandem Topology

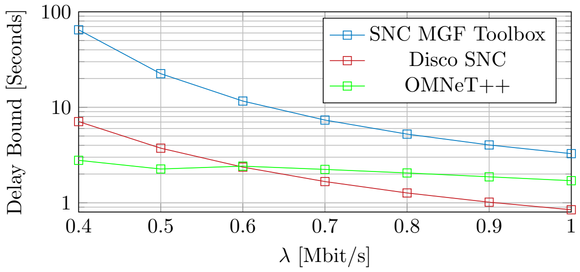

Figure 4: Delay Bound - Tandem Topology

DISCO DNCv2 provides, depending on the choice of θ, a bound of either infinity or 0; this is caused by the lack of burstiness of the arrival curve and latency of the service curves, which cannot be represented in the other programs. The results produced by DISCO DNCv2 are, therefore, not depicted.

3.3.4. OMNeT++. The variable "interArrivalTime" of the source refers to the time between generated jobs and is set to exponential(λ). The variable "serviceTime" refers to the time required by the queue to serve a job and is set to is $\frac{1}{5}$, $\frac{1}{4.9}$, and $\frac{1}{4.5}$ for the three queues respectively. The variable which represents the backlog bound for the entire system is the sum of the variable "queueLength:max" for all queues. The delay bound is represented by the variable "lifeTime:max" of the sink. Each simulation was run for 2,000,000 events.

3.4. Results & Analysis

Figures 3 and 4 depict, respectively, the backlog and delay bounds of the calculations and simulations for the tandem topology. Figures 5 and 6 depict, respectively, the backlog and delay bounds of the calculations and simulations for the tree topology.

The data, produced by the most uniform configuration of the heterogeneous tools possible, shows that the values for both the backlog and delay are similar for less congested networks, i.e. networks with larger λ values. In more congested networks, however, DISCO SNC almost always provides significantly tighter bounds. The values produced by the OMNeT++ simulations are frequently smaller than those calculated by the other tools as it does not necessarily provide a worst-case bound (with certain tolerance) but the worst-case values found in simulations.

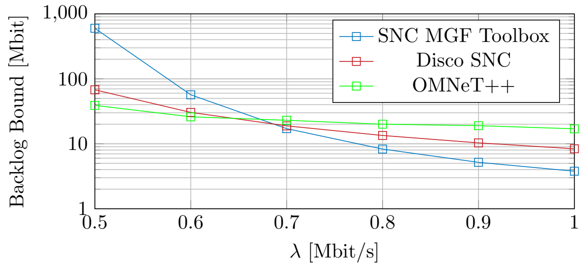

Figure 5: Backlog Bound - Tree Topology

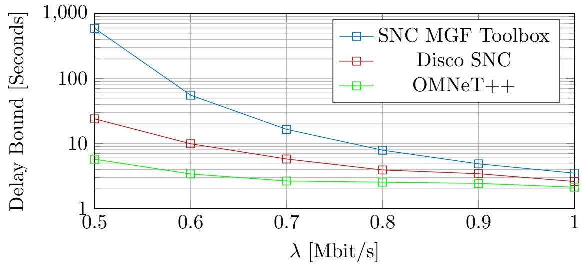

Figure 6: Delay Bound - Tree Topology

4. Conclusion & Further Work

This paper described the basics of stochastic network calculus as well as its similarities and differences with deterministic network calculus and queuing theory. Four tools used for the analysis of computer networks based on these theories / methods were then introduced, compared, and used to analyze simple reference topologies. The tools' heterogeneity posed the challenge of finding configurations that produce comparable results. The results show that, although the tools provided similar results for the tested networks, stochastic network calculus, specifically the DISCO SNC tool, could provide the tightest worst-case bounds. However, the limitation of the arbitrary scheduler available in the SNC MGF Toolbox contributed to the higher bounds produced by that tool. The results produced by DISCO DNCv2 were, unfortunately, incomparable with those produced by the other tools.

This paper was limited to the relatively theoretical application of stochastic network calculus. Further work could examine the bounds produced by the introduced tools in more complex settings and compare those with results from simulations in different simulators or real-world experiments.

References

[1] D. Ferrari, "Client requirements for real-time communication services," *IEEE Communications Magazine*, vol. 28, pp. 65–72, 1990.

[2] R. L. Cruz, "A calculus for network delay. I. Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, 1991.

[3] ——, "A calculus for network delay. II. Network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132–141, 1991.

[4] Y. Jiang and Y. Liu, *Stochastic Network Calculus*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[5] M. Fidler and A. Rizk, "A guide to the stochastic network calculus," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 92–105, 2015.

doi: 10.2313/NET-2020-11-1_16

[6] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat, “Synchronization and linearity - an algebra for discrete event systems,” *The Journal of the Operational Research Society*, vol. 45, 01 1994.

[7] A. Erlang, “The theory of probabilities and telephone conversations,” *Nyt Tidsskrift for Matematik B*, vol. 20, pp. 33–39, 01 1909.

[8] M. A. Beck and J. B. Schmitt, “The DISCO stochastic network calculator version 1.0 – when waiting comes to an end,” in *Proc. of the International Conference on Performance Evaluation Methodologies and Tools*, ser. ValueTools ’13, December 2013, pp. 282–285. [Online]. Available: https://dl.acm.org/citation.cfm?id=2631878

[9] P. Nikolaus, “Snc mgf toolbox.” [Online]. Available: https://github.com/paulnikolaus/snc-mgf-toolbox

[10] S. Bondorf and J. B. Schmitt, “The DiscoDNC v2 – a comprehensive tool for deterministic network calculus,” in *Proc. of the International Conference on Performance Evaluation Methodologies and Tools*, ser. ValueTools ’14, December 2014, pp. 44–49. [Online]. Available: https://dl.acm.org/citation.cfm?id=2747659

[11] A. Varga and R. Hornig, “An overview of the omnet++ simulation environment,” in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools ?08. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[12] M. A. Beck. [Online]. Available: https://disco.cs.uni-kl.de/discofiles/projects/SNC-Course/Script.pdf

doi: 10.2313/NET-2020-11-1_16