The DISCO Network Calculator

Nicos Gollan, Frank A. Zdarsky, Ivan Martinovic, Jens B. Schmitt

Distributed Computer Systems Lab (DISCO)
Technische Universität Kaiserslautern
p.o. box 3049
67653 Kaiserslautern, Germany

1 Network Calculus

Network calculus [1, 2] was developed for use in IP and ATM networks. It aims to be a system theory for deterministic queuing, allowing to derive deterministic guarantees on throughput and delay, as well as find bounds on buffer sizes and thereby allow for loss-free transfer. Traditional queuing theory analyzes the average or equilibrium behavior of a network, whereas network calculus shows the worst observable case. It describes network traffic and services in terms of *curves,* wide-sense increasing functions that represent accumulated data. By using a min-plus algebra, replacing the addition and multiplication in $(\mathbb{R}, +, \cdot)$ by minimum and addition $(\mathbb{R}, \min, +)$, it avoids non-linearities in the required operations on those curves. Similar to traditional system theory, network calculus provides a *concatenation result,* thereby making it possible to reduce tandems of systems (respective services) to a single equivalent service.

An extension of the network calculus, the *sensor network calculus* [3], provides results for handling cross-traffic in networks with tree topologies that are typical for sensor networks. More recent results handle effects like *data scaling,* which is an effect in networks that not only forward data, but also mutate it along the path. This may be the result of compression, aggregation, or other processing.

2 The Network Calculator

First presented in 2006 [4], the DISCO Network Calculator provides a framework to model and analyze networks with a number of methods. Implemented in Java, it runs on any platform with a Java run-time environment version 1.4 or higher[1]. This allows for a wide range of applications, from simple demonstrations, even in Java Applets, to complex solutions for large-scale problems.

The current release version including full source code and some examples can be downloaded at [5]. It allows to model networks in Java code or load graphs from GraphML files or other formats supported by the JUNG library [6]. Several analysis methods are provided, ranging from calculating the Charny Bound of a network and performing Fair Queuing Analysis, to three different methods from [7] (Total Flow, Separated Flow, and Pay Multiplexing Only Once Analysis), which are mainly focused

[1] Future versions of the Network Calculator will depend on features from Java version 1.5.

on analyzing sink tree networks[2], a common structure in wireless sensor networks. Depending on the analysis method, it is possible to perform the analysis with or without assuming FIFO service. It is also possible to refine service specifications by specifying *maximum service curves,* which represent an upper bound on the service available during a time span.

The current release is restricted to wide-sense increasing curves for both arrival and service curves and only allows "ultimately affine" functions that end in an infinite linear segment. Implementing the algorithms from [8] shows promising results by lifting that restriction. The authors present a computational framework for general piecewise-affine functions, which allows to perform calculations on more complex curves resulting from TDMA or other discrete medium access schemes. The results from using those curves within the Network Calculator yield results very close to those found in simulations [9].

Using the Network Calculator mainly involves three steps. First, a model of the network has to be acquired. Currently, this means modeling the network in a tool that can export to a description language understood by the JUNG library, or generating the model from Java code. Additionally, it is possible to use topology generators like BRITE [10] together with an output plug-in available at [5]. In a second step, the basic input- or forwarding behavior of each node is described in the form of arrival and service curves. The third step is to analyze the complete model using one of the supported analysis methods, resulting in a delay- or buffer bound for a given flow of interest through the network.

3 Examples of Use

The DISCO Network Calculator has been used in our research and yielded good results. Examples can be seen in [9, 11–13]. In [11], the tool was used to find the solution to an optimization problem, whereby optimal parameters for a TDMA scheme were determined. Results from [9] show that the bounds calculated with network calculus are close to simulation results for a sensor network. Consequently, a system designed from network calculus results is not necessarily over-dimensioned.

4 Future Work

We plan to extend the Network Calculator by creating a graphical user interface to make network creation and scenario modeling easier. Currently, those processes require the user to write Java code and are disjoint due to the lack of a storage format supporting curve specifications. In the process, we will also create graphical editors for network graphs, as well as service and arrival curves. Furthermore, a description language to easily import and export annotated network models is needed; a candidate would be an extended GraphML dialect. By extending the Network Calculator to have more functionality and be more user-friendly, we hope to open it up to a wider audience. Ultimately, using such tools may become an integral part of mission-critical network design.

[2] Sink tree networks are structured as trees, with input being generated at the nodes and forwarded all the way to the sink which is located at the root.

References

1. Rene L. Cruz. A Calculus for Network Delay, Part I+II. *IEEE Transactions on Information Theory*, 37(1):114–141, 1991.
2. Jean-Yves Le Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, volume 2050 of *Lecture Notes in Computer Science*. Springer, 2001.
3. Jens Schmitt and Utz Roedig. Sensor Network Calculus - A Framework for Worst Case Analysis. In *Proceedings of IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS'05), Marina del Rey, USA*, pages 141–154. Springer, LNCS 3560, June 2005. ISBN 3-540-26422-1.
4. Jens B. Schmitt and Frank A. Zdarsky. The DISCO Network Calculator - A Toolbox for Worst Case Analysis. In *Proceedings of the First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'06), Pisa, Italy*. ACM, November 2006.
5. The DISCO Network Calculator. `http://disco.informatik.uni-kl.de/content/Downloads`, January 2008.
6. JUNG – the Java Universal Network/Graph Framework. `http://jung.sourceforge.net/`, January 2008.
7. Jens B. Schmitt, Frank A. Zdarsky, and Ivan Martinovic. Performance Bounds in Feed-Forward Networks under Blind Multiplexing. Technical Report 349/06, University of Kaiserslautern, Germany, April 2006.
8. Anne Bouillard and Éric Thierry. An Algorithmic Toolbox for Network Calculus. Technical report, Unité de recherche INRIA Rennes, 2007.
9. Utz Roedig, Nicos Gollan, and Jens Schmitt. Validating the Sensor Network Calculus by Simulations. In *Proceedings of the 2nd Performance Control in Wireless Sensor Networks Workshop at the 2007 WICON Conference, Austin, Texas, USA*. ACM Press, October 2007.
10. Boston University Representative Internet Topology Generator (BRITE). `http://www.cs.bu.edu/brite/`, January 2008.
11. Nicos Gollan and Jens B. Schmitt. Energy-Efficient TDMA Design Under Real-Time Constraints in Wireless Sensor Networks. In *15th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'07)*. IEEE, October 2007. Accepted for publication.
12. Wint Yi Poe and Jens B. Schmitt. Minimizing the Maximum Delay in Wireless Sensor Networks by Intelligent Sink Placement. Technical Report 362/07, University of Kaiserslautern, Germany, July 2007.
13. Jens B. Schmitt, Frank A. Zdarsky, and Markus Fidler. Delay Bounds under Arbitrary Multiplexing: When Network Calculus Leaves You in the Lurch ... In *27th IEEE International Conference on Computer Communications (INFOCOM 2008)*, Phoenix, AZ, USA, April 2008.