Rare Events in Network Simulation Using MIP

Emanuel Heidinger
EADS Innovation Works, Germany

***Abstract*—Switched Ethernet has become a serious candidate for real-time communication in industry and consumer electronics. Due to the unpredictable nature of switched networks, tools and techniques became necessary to determine worst case behaviour, such as network calculus, worst case scheduling analysis and network simulation. This paper proposes a network simulation technique to estimate the results expected from a deterministic network calculus analysis. Traffic will be generated from network calculus arrival curves. Packet generation will be triggered at points of time that provoke a worst case situation. These points will be identified by solving a mixed integer programming problem.**

I. Introduction

It is mandatory to address worst case behaviour in safety relevant switched Ethernet communication networks with analytical tools such as the network calculus (NC). NC was basically formed in the 90ies by Cruz [1] [2] and Le Boudec [3] and is primarily designed to determine worst case transmission and queuing delays, i.e., the delays that are characterized by heavy variability, which is inherently more difficult than determining the relatively stable values of propagation and processing delay. However, recent work has shown that NC is usually not able to identify tight bounds using techniques from NC such as Total Flow Analysis (TFA), Separated Flow Analysis (SFA) or Pay Multiplexing Only Once SFA (PMOO-SFA) [4]. Recent work [4], [5] has introduced techniques to determine tight bounds solving linear optimization problems, where [5] has also shown, that finding such tight bounds is NP-hard for the general case and even for the simpler case of feed forward networks.

The discrete event simulation of the communication behaviour using Monte Carlo methods is another, accepted method to estimate delay, utilization and backlog especially from larger networks. Within a simplified view, the simulation scenario is built up of traffic generators and receivers, network nodes as well as data links. The standard models to generate traffic flows include Markov on-off processes, distributions over packet inter-arrival times and sizes, as well as traffic traces from real-world sources to name a few. See [7] for a list of references.

The network nodes – in particular switches and multiplexer – require a realistic mapping of the queuing system, scheduling algorithms and scheduling strategies. The accuracy of the simulation depends on the quality of the employed traffic and network models, which implies that a simulation of worst case scenarios is not straightforward and requires careful mapping. In fact finding such rare events regularly requires long-running simulations.

In this work we implement network models and traffic generators to determine the end-to-end worst case delay and the induced backlogs. Traffic is generated from NC arrival curves. In order to provoke the rare worst case situation in the simulation, the traffic generating processes have to be aligned to each other. This alignment will be determined by solving a mixed integer program (MIP). Traffic is characterized by the token bucket traffic model [6], which is common in the field of NC and which can be enforced by shapers implemented in hardware.

This paper provides the following key contributions:

- We propose the employment of a mixed integer program that represents topology and traffic flows. The solution provides the start points to the traffic generators that provoke worst case queuing in the model.
- Compared to regular network simulation, this kind of rare event simulation provokes worst case queueing at the very first rounds achieving enormous speed-up.
- The translation to a mixed integer program allows capturing the exponential increase of queuing orders elegantly. This strategy requires solving one single mixed integer program.

Related work that already aimed at a combination of NC and network simulation was achieved by Kim et al. [8]. They proposed an automated hybrid approach of simulation and NC to speed up network simulations. Our approach tries to shift bounds from a network simulation towards tight upper bounds.

II. From Arrival Curves to Mixed Integer Program

Without loss of generality – the introduced mixed integer program problem can also be stated for general feed forward networks – we consider the simple tandem scenario illustrated in Figure 1, which is similar to the one found in [4]. The traversing flows $F1$ – $F4$ follow the token bucket model with the parameters listed in Table I.

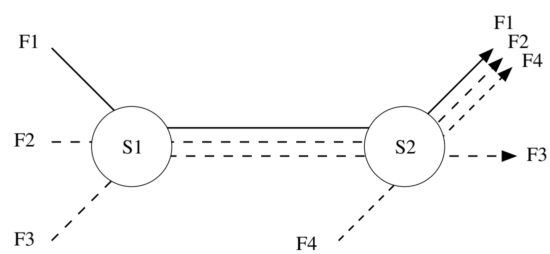

Fig. 1. Tandem scenario with four flows

To identify the alignment of the start points that cause a worst case situation, we formulate a mixed integer program,

978-0-9836283-0-9

This paper was peer reviewed by subject matter experts for publication in the Proceedings of ITC 2011

Flow	Burst [bytes]	Rate [kBits/s]
$F1$	108	100
$F2$	64	1000
$F3$	256	2000
$F4$	512	5000

TABLE I
TOKEN BUCKET PARAMETERS

such that all possible queuing orders can also be expressed within the same mixed integer program as well. For the later considerations, one unit corresponds to the transmission time of a single byte, e.g., $0.08\mu s$ in case of Fast Ethernet. Due to employment of (single) token bucket models, we do not have to consider token rates in our mixed integer program, since all traffic occurs well-shaped, apart from the occurring burst already paid. Other traffic models as the dual token bucket model or those common in network simulation (see above) will be addressed in future work.

The mixed integer program is solved using a standard MIP solver. For the addressed worst case scenario, where the delay of flow $F1$ should be maximized, the MIP solver states the following results: Worst case generation point of $F1$ is at 300, $F2$ at 344, $F3$ at 152 and $F4$ at 340. The maximum delay due to queueing effects is 1546, which corresponds to $123.86\mu s$ in case of Fast Ethernet.

III. Preliminary Evaluation

In fact, running the simulation with the start points found in the last section, already provokes the worst case delay as observed in all network simulations carried out. The preliminary evaluation goes a step beyond and investigates the area of the worst case point by varying each start point by $n \cdot 2.4\mu s$, where $-2 \leq n \leq 2$. The start point of this iterative process is the worst case as defined in the last section, i.e., $(300, 344, 152, 340)$. The network simulation is performed with the network simulator OPNET [9].

Figure 2 compares the results from the mixed integer program with the results from the network simulation, which are higher since we omitted delays from protocol stacks in the mixed integer program. The red bar illustrates the start point $(300, 344, 152, 340)$. As a result, the mixed integer program was able to predict most results from the simulation. While developing a more sophisticated version of this mixed integer program, we get first hints on the mismatches seen in comparison, which apparently arise from an unrealistic mapping of network components to the mixed integer program concerning processing delay, propagation delay and delays from protocol stacks. One point seems to be be critical: The worst case seen in the mixed integer program causes two different delays in the simulation with only one being the exact worst case. However, with respect to rare event simulation, provoking worst case queueing using start points from the mixed integer program allows speeding up the network simulation since the worst case will be observed in the very first seconds.

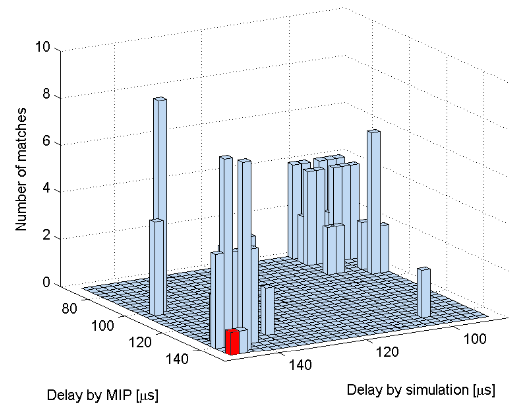

Fig. 2. Integer program versus simulation

IV. Conclusion

In this paper, we proposed to translate the network topology, i.e., service curves and arrival curves to a mixed integer program. This hard optimization problem is solved using an optimized MIP solver. Compared to previous work, we capture queuing elegantly while moving to an NP-hard problem. In fact, finding tight bounds is already known to be NP-hard.

With the combination of techniques from the fields of network calculus and network simulation, we approximate the worst case bound from the bottom up. We always achieve performance bounds that were observed in the simulation, moving us forward towards tighter bounds in simulation.

Furthermore, we have shown the basic forecast of worst case situations in a simple tandem scenario. Future work will address more complex topologies as well as parameter studies regarding token bucket parameters, processing time and minimum inter-frame gap to achieve a better overlap.

References

[1] R.L. Cruz. A Calculus for Network Delay, Part I, Network Elements in Isolation. *Information Theory, IEEE Transactions on*, 37(1):114 –131, jan. 1991.

[2] R.L. Cruz. A Calculus for Network Delay, Part II, Network Analysis. *Information Theory, IEEE Transactions on*, 37(1):132 –141, jan. 1991.

[3] J.Y. Le Boudec. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. 2004.

[4] J.B. Schmitt, F.A. Zdarsky, and M. Fidler. Delay Bounds under Arbitrary Multiplexing: When Network Calculus Leaves You in the Lurch... In *IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, pages 1669–1677, 2008.

[5] A. Bouillard, L. Jouhet, and E. Thierry. Tight performance bounds in the worst-case analysis of feed-forward networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[6] S. Shenker and J. Wroclawski. General characterization parameters for integrated service network elements, 1997.

[7] M. Fidler. A Survey of Deterministic and Stochastic Service Curve Models in the Network Calculus. *Communications Surveys Tutorials, IEEE*, 12(1):59 –86, 2010.

[8] H. Kim and J.C. Hou. Network Calculus Based Simulation for TCP Congestion Control: Theorems, Implementation and Evaluation. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2844–2855. IEEE, 2004.

[9] OPNET Technologies, Inc. Application and Network Performance with OPNET. Website, Accessed July 4, 2011. http://www.opnet.com/.