Array-based SMT Encoding for Congestion Control Algorithms

June 19, 2023

Problem description

Recently, there has been an increased interest from the research community in using formal methods for analyzing network performance [1, 2]. The proposed techniques are based on SMT solvers (such as Z3 [7]), can model network queues [1] or various congestion control algorithms [2], and can automatically synthesize counter-examples that expose poor performance (e.g., packet loss, low utilization, unfairness). However, they encode continuous models in finite, discrete time, as these representations can be solved more efficiently. These approximations may thus lead to false positives and do not allow one to prove that the desired property holds over larger (potentially infinite) time horizons.

The goal of this project is to design a more generic SMT encoding for congestion control algorithms, which addresses the limitations mentioned above. In particular, we will focus on encoding operations over queues and cumulative functions as operations over arrays, since arrays in SMT-LIB [4] are unbounded and can have indexes of arbitrary types[1].

Example

Let us consider the leaky bucket algorithm, where the packets received by a server are stored in a temporary buffer ("bucket") and then sent at a constant rate. The cumulative number of received packets is given through the arrival curve $\alpha(t) = a * t + b$, where a is the arrival rate and b is the burst tolerance. Following the ideas from [1, 2], the arrival curve can be encoded in SMT as a sequence of real variables $\alpha_1, ..., \alpha_T$, where T is a constant, representing the maximum number of time steps and $\alpha_i = a * i + b$.

However, one can also use a more generic encoding based on arrays:

$\forall t \in \mathbb{R} : select(\alpha, t) = a * t + b$ (1)

where α is an SMT-LIB array with real indexes and real values and *select* is the SMT-LIB function for accessing an element of the array at a given index.

[1] *http://smtlib.cs.uiowa.edu/theories-ArraysEx.shtml*

Approach

We will start with the SMT encodings generated by the artifact[2] of [2] for the congestion control algorithms considered in their evaluation: AIMD [6], BBR [5], and Copa [3]. We will then modify the encodings to use arrays (and potentially uninterpreted functions), instead of unrolling the queues and the cumulative functions over time.

Note that our array-based encodings may contain universal quantifiers (as shown in (1)), and quantifier instantiation is generally undecidable. The SMT solver might thus also return *unknown* (not only *sat, unsat*, and *timeout* as before). In such cases, we may also need to augment the encodings with additional terms that trigger particular quantifier instantiations and enable the solver to prove/disprove the satisfiability of the given formula.

At the implementation level, we will extend the artifact to also automatically generate our new encodings and store them in files. We will then automatically run the SMT solver Z3 from command line and integrate its result with the existing visualizer for displaying counter-examples.

Prerequisites

The student is expected to have good programming skills. Prior experience with the SMT-LIB format is a plus.

Opportunities

The student will have the chance to gain a deep understanding of SMT solvers and to learn about congestion control algorithms.

Contact

Alexandra Bugariu: bugariua@mpi-sws.org

References

[1] Mina Tahmasbi Arashloo, Ryan Beckett, and Rachit Agarwal. Formal methods for network performance analysis. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 645–661, Boston, MA, April 2023. USENIX Association.

[2] Venkat Arun, Mina Tahmasbi Arashloo, Ahmed Saeed, Mohammad Alizadeh, and Hari Balakrishnan. Toward formally verifying congestion control behavior. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*,

[2] https://github.com/venkatarun95/ccac

SIGCOMM '21, pages 1–16, New York, NY, USA, 2021. Association for Computing Machinery.

[3] Venkat Arun and Hari Balakrishnan. Copa: Practical Delay-Based congestion control for the internet. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 329–342, Renton, WA, April 2018. USENIX Association.

[4] Clark Barrett, Pascal Fontaine, and Cesare Tinelli. The Satisfiability Modulo Theories Library (SMT-LIB). `www.SMT-LIB.org`, 2016.

[5] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. Bbr: Congestion-based congestion control. *Commun. ACM*, 60(2):58–66, jan 2017.

[6] Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17(1):1–14, 1989.

[7] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'08/ETAPS'08, pages 337–340, Berlin, Heidelberg, 2008. Springer-Verlag.