A Residual Service Curve of Rate-Latency Server Used by Sporadic Flows Computable in Quadratic Time for Network Calculus (Artifact)

Marc Boyer
ONERA / DTIS – Université de Toulouse, F-31055 Toulouse, France

Pierre Roux
ONERA / DTIS – Université de Toulouse, F-31055 Toulouse, France

Hugo Daigmorte
RealTime-at-Work, F-54600 Villers-lès-Nancy, France

David Puechmaille
RealTime-at-Work, F-54600 Villers-lès-Nancy, France

Abstract

Computing response times for resources shared by periodic workloads (tasks or data flows) can be very time consuming as it depends on the least common multiple of the periods. In a previous study, a quadratic algorithm was provided to upper bound the response time of a set of periodic tasks with a fixed-priority scheduling. The related paper generalises this result by considering a rate-latency server and sporadic workloads and gives a response time and residual curve that can be used in other contexts. It also provides a formal proof in the Coq language. This artifact enables to reproduce this proof.

2012 ACM Subject Classification Networks → Formal specifications; Networks → Network performance evaluation; Networks → Network reliability; Software and its engineering → Formal methods; General and reference → Verification

Keywords and phrases Network Calculus, response time, residual curve, rate-latency server, sporadic workload, formal proof, Coq

Digital Object Identifier 10.4230/DARTS.7.1.2

Related Article Marc Boyer, Pierre Roux, Hugo Daigmorte, and David Puechmaille, "A Residual Service Curve of Rate-Latency Server Used by Sporadic Flows Computable in Quadratic Time for Network Calculus", in 33rd Euromicro Conference on Real-Time Systems (ECRTS 2021), LIPIcs, Vol. 196, pp. 14:1–14:21, 2021. `https://doi.org/10.4230/LIPIcs.ECRTS.2021.14`

Related Conference 33rd Euromicro Conference on Real-Time Systems (ECRTS 2021), July 5–9, 2021, Virtual Conference

1 Scope

This artifact enables to reproduce the formal proof of the related paper.

2 Content

The artifact package includes:

- a `README.md`;
- a `theories/` folder containing the source code of the Coq formal proof (files with the `.v` extension);
- a `Makefile` relying on a `_CoqProject` file to help build the proof.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS).

4 Tested platforms

The artifact requires a system able to run the Coq software.

5 License

The artifact is available under license GPL-2.0.

6 MD5 sum of the artifact

c532fe1a3cd11a828d7891b07b707363

7 Size of the artifact

15 kiB

A Artifact evaluation

The provided artifact enables to reproduce the formal proof of Section 4.2 of the related paper[1].

A.1 Virtual Machine

A virtual machine with all required softwares already installed is provided at `https://zenodo.org/record/4734308` with instructions in its file `/home/ecrts21/Coq/README.md`. If need be, the login/password of the virtual machine are `ecrts21/ecrts21`. The remaining alternatively details installation and compilation on a fresh system.

A.2 Installation and compilation on a fresh system

A.2.1 Prerequisites

This needs:

- Coq (tested with version 8.13.0)
- the MathComp library (components ssreflect and algebra, tested with version 1.12.0)
- the Mathcomp analysis library (tested with version 0.3.5)
- the Coquelicot library (tested with version 3.2.0)
- the Flocq library (tested with version 3.4.0)

[1] The scripts for reproducing the evaluation of Section 5 can unfortunately not be shared as they rely on some proprietary software.

To install all those dependencies, the easisest solution is to use the package manager OPAM `https://opam.ocaml.org/`:

```
% opam repo add coq-released https://coq.inria.fr/opam/released
% opam update
% opam install coq.8.13.0 coq-mathcomp-algebra.1.12.0 \
    coq-mathcomp-analysis.0.3.5 coq-coquelicot.3.2.0 coq-flocq.3.4.0
```

This should take a few dozen minutes. Note that you'll need OPAM 2 (old OPAM 1 won't work). This may require a few system dependencies, for instance the developement files of the GMP library which can be installed by `apt-get install libgmp-dev` on Debian based systems, before re-running `opam install ...`.

A.2.2 Compilation

Once above prerequisites are installed, just type make:

```
% make
```

The proof succeeds when there is no errors and a `file.vo` file is produced for each `file.v` in `theories/`. One can also check for the absence of additional axioms in the code (presence of the keywords `Axiom`, `Parameter` or `Admitted`).

A.2.3 Documentation

To generate the documentation:

```
% make doc
```

You can then open `html/toc.html` with your favorite browser or more precisely `html/qbound.quadratic_bound.html` for the proof of the main theorem.