*“An experimental result is not fully established unless it can be independently reproduced”**

*https://www.acm.org/publications/policies/artifact-review-badging

Workshop objective

- Craft recommendations on how to
  - generalise the reproducibility
  - review research results and artifacts
- for publications in the SIGCOMM interest group.

Terminology

- Repeatability
  - Same team, same experimental setup
- Replicability
  - Different team, same experimental setup
- Reproducibility
  - Different team, different experimental setup

Badging

- Artefact available: publicly archived
- Artefact evaluated
  - Functional: Documented, Consistent, Complete, Exercisable

  - Reusable: functional + well documented and easy reuse/ repurposing

- Artefact validated
  - Replicated

  - Reproduced

Reviewing process I/O

- 11 submissions
- 7 accepted papers
  - 4 shepherd
- reviewed by 15+3 devoted reviewers from the TPC
  - min/median/avg/max = 4/5/5.09/7
- 2 CoNEXT’16 testimonials on the wiki.

Program of the day

- 9:00am - 9:15am
  Opening Remarks

- 9:15am - 10:30am
  Session 1 - *Why is reproducibility so hard?*

- 10:30am - 11:00am
  Coffee Break (Foyer)

- 11:00am - 12:30pm
  Session 2 - *How reproducible is our research?*

- 12:30pm - 2:00pm Lunch Break (Centennial Terrace)

- 2:00pm-3:30pm
  Group discussions

- 3:30pm - 4:00pm
  Coffee Break (Foyer)

- 4:00pm - 4:45pm
  Work groups wrap-up and conclusions

- 4:45pm - 5:25pm
  reporting

- 5:25pm - 5:30pm
  Closing remarks

CoNEXT'16

Testimonials

HyPer4: Using P4 to Virtualize the Programmable Data Plane

“My primary concern was to provide access to the codebase. So I provided a link to it in the paper. This included the scripts and utilities I used for evaluation along with the code of the research system itself. Unfortunately, soon after submitting the paper, the codebase changed fundamentally such that many of the scenarios discussed in the paper no longer work. Because it is a git repository, however, the working version is still available - but 1) nobody but me knows the specific git commit to get the working version; and 2) even then, numerous tweaks required to make it work were not captured by that commit for some reason. We could add 3) no doubt there are many external dependencies that must be satisfied before the environment can support it. Packaging all of this up in some kind of container or VM would probably be the way to go. I didn't prioritize this packaging at the time I submitted the paper, reasoning that I would continue down the research path now, and respond to demand for help reproducing my results, or demand for help just using my research system, when such demand showed up later, if it ever did.

I have since created a branch of the repository that provides the correct version of the codebase that can show that the system works as claimed, and am in the process of documenting the commands to run, known dependencies, and so forth. While this complete support for reproducibility may be a bit late, at least the system code has always been available, flawed as it may be, for learning and critique, by anyone who has the paper.”

David Hancock

LossRadar: Fast Detection of Lost Packets in Data Center Networks

“I am the first author of this paper. The evaluation part has both simulation and testbed evaluation. For the simulation, our goal is to evaluate how LossRadar performs (%capture and accuracy) under certain traffic distribution. So we first generate the per-hop packet-level record in ns3, and then we apply LossRadar to test the accuracy. We have all the packet-level records, so it is easy to reproduce the result. We did not publish the records because they are >20GB, but we are happy to share them with anyone who wants to reproduce it.

For testbed, our goal is to evaluate the loss detection delay and how it performs in improving flow throughput. We built the system in Deterlab, and modified Open vSwitch to support LossRadar. We've done the experiment 2 years ago, and submitted before. After the submission I started another project that has similar architecture with LossRadar, so I directly modified the OVS code. I did not use Github at that time, so the original code for LossRadar is lost. I am happy to help anyone who wants to reproduce the result, and tell him/her the details of my implementation in my mind.

Improving reproducibility is an important thing. I think one thing to keep in mind is that to always do the version control, and make separate code bases for different projects.”

Yuliang Li

Backup

set of questions

Why is it hard to get it?

- lack of non-ambiguous terminology
- what are the artefact to make available, how to describe them, what version ?
    - dataset
    - script (comment bien faire pour les utiliser)
    - infrastructure
    - expiration of URI
- reviewing process
    - time consuming
    - **how to ensure double-blind**
    - good reviewing work not really valued
- **lack of incentives (idea are the most valued)**
- **papers are punctual (authors do something after)**

How to

- a reproducibility challenge/contest
- systems to upload artefacts
- meta-artefact to describe artefacts
- have a reproducibility discussion section (materials and methods)
- include reproducibility questions in reviewing forms
- highlight (how to badge/evaluate)
- shared evaluation environments
    - testbeds, dataset, workload
- systematiquement faire un editorial pour les conf
- faire étude des papiers les plus cités et regarder proba d’artefacts

- how to define we are “equal” -> class of equivalence