

Session 4: Session 4: Digital Twinning the Network - can we perfect it?

Transcribed by: M. Taimoor Tariq, Ammar Tahir (UIUC)

Generating representative, live network traffic out of millions of code repositories

Tobias Bühler, Roland Schmid, Sandro Lutz, Laurent Vanbever (ETH Zürich)

There are limited options for network data today, we can use Github repository to generate live traffic by running open source projects. Use automation frameworks like docker to easily run these open source projects. Through Dynamo, run code repos and generate traffic to learn classification of traffic.

Questions:

1. How much gold is there actually?

We do have a low success rate since there is a lot of irrelevant code. Processing the readme alone has allowed us to increase our success rate. In the future, we want to scale to other techniques such as vagrant, kubernetes. I hope that we can automate the entire process and get the relevant gold mine data out.

2. How do you detect the categorization of traffic?

Mostly on the basis of paths taken by the traffic.

3. Variations in coding styles, so many user lists. How to go about it?

Used a Github Search API. Filtered out repositories which were not of interest. You can also use metadata to check for the correctness of/relevance of data.

4. What about the correctness of generated traffic?

Right now we focus on generating traffic, future work can look into classifying traffic as correct or not.

SEED Emulator: An Internet Emulator for Research and Education

Wenliang Du, Honghao Zeng, Kyungrok Won (Syracuse University)

Seed is a Python based Internet Emulator which has the ability to emulate many things from ASes for routing to protocols and applications like Ethereum. Really awesome demo!

Questions:

1. Are you using actual application (Ethereum) code? How do you simulate real world behavior?

Yes! And right now its random transactions from random users. In future work, we will try to use real world examples instead.

2. What software stack of BGP are you running?

We are using the Open source BGP code, Bird.

The Internet of Things in a Laptop: Rapid Prototyping for IoT Applications with Digibox
Silvery Fu, Hong Zhang, Sylvia Ratnasamy, Ion Stoica (UC Berkeley)

Author presented Digibox which is a simulator for an IoT system. This simulator allows a scene based simulation where along with devices, the environment and its interaction with IoT devices can also be modeled.

Questions:

1. Do you have a physical simulation?

It is Software based but devices are simulated based on vendor specification of devices.

2. How to coordinate across devices?

That can be done in the simulation.

3. Create a game engine with something like unity in order to allow for interaction.

That would be quite useful, we are going to explore this in future work.

End of Session Discussion

Venkat Arun from MIT: A lot of people develop simulators for different things. How feasible is it to combine different simulators from different things to make one multi purpose simulator.

It can work but its a nightmare. Not easy to achieve.

How the software for web scraping github would depend alot on the context it is being run in? How do we make simulations (future proof?)

For now we are just trying to make the system work, then we want to streamline it based on apps. For example if we have a DB application, if you are not making good DB queries, the results would reflect that and you can interpret that context as missing.

E.g. how do i use the system (IoT Silvery Fu) for monitoring on a farm

Instead of making future proof simulations, we should focus on enabling agility and ease to design and test new solutions now.

Would the SEED emulator allow multiple people/devices to connect? Developing these together and developing together in the real world.

For SEED, yes you can sign in and attach and participate in the emulation.

We have been working on a feature similar to this where data can be sent across different testbeds.

Run and simulate a code or have a real setting with actual (probably harder) implementations. How long would it take to implement a realistic BGP implementation.

Not sure how small it would be but it would be an interesting educational approach.

These existing systems probably require niche languages. Generalized it a natural/higher level language which can automatically map it to different specification, e.g. through NLP

It might be challenging to come up with one which is broad enough to cover all use cases but also cover all the broad cases we have.

Yes, very useful. Should be made. The SEED emulator was able to make it work with Python.
