Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prof. Laurent Vanbever
Networked Systems Group

A framework for collecting data traffic from real networks

Semester thesis proposal

Computer network researchers often need to evaluate a new system on real traffic data to assess whether it works in practice. The problem though is that getting suitable real traffic traces is not always simple. While some organizations [2, 4] make publicly available traffic traces captured in real networks, their number and duration are very limited, they are also usually captured from few links only. Consequently, researchers often end up evaluating their ideas on few traces (at best), such as the ones provided by CAIDA [2].

In this thesis, the goal is to develop a framework to collect traffic traces in real time, and that operators can deploy in their network and use to later make the traces available to the community. Previous works already implemented similar frameworks [8–12], but some do not scale to high traffic rate, some others are hard to deploy in real networks, or lack of flexibility. To (hopefully) convince network operators to deploy the framework, it must *i)* be easy to set up for the operators, *ii)* guarantee users privacy by anonymizing the traffic traces (*e.g.,* CAIDA uses a prefix preserving anonymization [3]), and *iii)* be flexible as network operators may have different privacy policies.

To implement the framework, the student can rely on the recent advances in network programmability, and more particularly on the recent programmable switches [7] (such as Tofino [1]). These programmable switches can forward packets at Tbps, and their hardware pipeline can be programmed using P4 [6]. In this theses, the idea is to use these programmable switches to perform the anonymization of the data traffic directly in the data plane, before sending the anonymized data to an external server where further processing can be performed (*e.g.,* data storage). The framework must support different levels of anonymization, and operators would only have to specify the anonymization in a high level language (similarly to [11]). The student will also have to design the framework so that it can handle potentially Tbps of traffic (often seen in real networks). For instance, the framework should be able to filter some traffic (*e.g.,* only focus on some destination prefixes, or protocols), or compress data traffic on-the-fly.

Milestones

- Understand the problem and the challenges, and study the existing solutions;
- Implement a P4 program that can anonymize traffic in the data plane according to operators-provided policies. The P4 program can be tested using a software switch such as BMv2 [5];
- Implement a program that runs on the server collecting the anonymized traffic. This program must for instance take care of storing the data;
- Test the full framework on a Tofino switch that we have in our lab.

Prerequisites

- Being able to program in Python, good knowledge in UNIX-like systems, some knowledge in P4 is a plus;
- Communication Networks (227-0120-00L), or equivalents.

Contact

- Thomas Holterbach, thomahol@ethz.ch
- Prof. Dr. Laurent Vanbever, lvanbever@ethz.ch

References

[1] Barefoot tofino, the world's fastest p4-programmable ethernet switch asics. https://barefootnetworks.com/products/brief-tofino/.

[2] The caida anonymized internet traces dataset. http://www.caida.org/data/passive/passive_dataset.xml.

[3] Cryptography-based prefix-preserving anonymization. https://www.cc.gatech.edu/computing/Networking/projects/cryptopan/.

[4] Mawi working group traffic archive. http://mawi.wide.ad.jp/mawi/.

[5] p4 behavioral model. https://github.com/p4lang/behavioral-model.

[6] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker. P4: Programming protocol-independent packet processors. *SIGCOMM Comput. Commun. Rev.*, 44(3):87–95, July 2014.

[7] P. Bosshart, G. Gibb, H.-S. Kim, G. Varghese, N. McKeown, M. Izzard, F. Mujica, and M. Horowitz. Forwarding metamorphosis: Fast programmable match-action processing in hardware for sdn. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 99–110, New York, NY, USA, 2013. ACM.

[8] A. Hussain, G. Bartlett, Y. Pryadkin, J. Heidemann, C. Papadopoulos, and J. Bannister. Experiences with a continuous network tracing infrastructure. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data*, MineNet '05, pages 185–190, New York, NY, USA, 2005. ACM.

[9] D. Koukis, S. Antonatos, D. Antoniades, E. P. Markatos, and P. Trimintzios. A generic anonymization framework for network traffic. In *2006 IEEE International Conference on Communications*, volume 5, pages 2302–2309, June 2006.

[10] J. C. Mogul and M. Arlitt. Sc2d: An alternative to trace anonymization. In *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, MineNet '06, pages 323–328, New York, NY, USA, 2006. ACM.

[11] R. Pang, M. Allman, V. Paxson, and J. Lee. The devil and packet trace anonymization. *SIGCOMM Comput. Commun. Rev.*, 36(1):29–38, Jan. 2006.

[12] S. Ubik, P. Zejdl, and J. Halak. Real-time anonymization in passive network monitoring. In *International Conference on Networking and Services (ICNS '07)*, pages 100–100, June 2007.