Verification of Programmable Networks

Dragoș Dumitrescu

Advisor: Prof. Dr. Ing. Dragoș Niculescu

Facultatea de Automatică și Calculatoare
Universitatea Politehnica București

This dissertation is submitted for the degree of
Doctor of Philosophy

Bucharest May 2022

Summary

Keywords: networks, model-checking, verification, P4, SDN, BGP, network verification, network correctness, cloud networking.

Computer networks are at the center of modern day lives, they are the building blocks of all compute systems. As such, they need to be highly reliable and bug-free to ensure the correct functioning of safety and mission critical applications. The advent of network data- and control-plane programmability as key enablers of multi-tenant cloud infrastructures has made networks more and more complex; it is thus increasingly difficult to manually assess their correctness. In order to tackle this problem, a new area of research – network verification – has seen significant success in the past decade.

In this dissertation, I will report on my research in making network verification algorithms practical in modern programmable networks. Practicality is explored in two flavors: ease of use and performance. In what ease of use is concerned, I explore implicit correctness properties which must hold for all networks under consideration, thus requiring a *zero-specification* effort from administrators. In what performance is concerned, the main requirement is to have verification tools scale reasonably well to large public cloud-provider networks, which may be comprised of thousands of routers, servers and network appliances.

The contributions of this thesis are positioned at the layers of programmability involved in a typical cloud network environment and show how my work tackles some of the hurdles encountered at each layer. Dataplane programming (e.g., P4) deals with programming forwarding devices; software defined networking (e.g. OpenFlow) handles the orchestration of multiple network devices while control plane configuration (e.g., routing policies) handles dissemination of control-plane information across the entire provider network. My research answers questions at all these layers. First of all, my contributions to Vera (Stoenescu et.al., *Debugging P4 programs with Vera* in SIGCOMM'18), Vera2 (Dumitrescu et.al. Dataplane verification for P4 presentation in NetPL'19) and `bf4` (Dumitrescu et.al., *bf4: Towards bug-free P4 programs* in SIGCOMM'20) show how detecting bugs in P4 programs can be made practical and easy to use. Secondly, `netdiff` (Dumitrescu et.al. *Equivalence and its applications to network dataplanes* in NSDI'19) is an algorithm which employs network dataplane equivalence to find

bugs in SDN controllers. My third contributions makes BGP simulation and model-checking efficient for production-scale cloud networks.

Contributions summary. The positioning of my work in the landscape of network verification hints at the prominent research endeavor tackled within my thesis: "finding efficient and scalable decision procedures in the context of complex and programmable network policies".

The most important principle my work is driven by consists of *ease of specification* or even *zero-specification*. My efforts in both programmable network dataplane and SDN controller verification use implicit notions of correctness and tailor decision algorithms based on their specifics.

My work in control plane verification builds upon tools which are innately customized checks meant to increase scalability. Thus, they obey the same previously advertised *zero-specification* principle. While this design decision is not part of my contribution, the approach is nonetheless worth mentioning as an alternative to existing forms of *zero-specification*. Given the properties under verification, my contributions focus on scaling BGP simulation algorithms to production networks. This is the second principle behind my work: building verification solutions which *scale to large networks*.

The main research question of this thesis can be summarized as follows: "Can one efficiently verify the correctness of programmable networks with little (preferably no) specification effort from network users?".

Note on previously published material. Some of the chapters in this thesis contain excerpts of already published scientific papers. The contents of sections 4.1 and 2.6.2.8 will undergo submission for publication in the near future.

Thesis outline. Starting from the lowest layer in the OSI model, in Chapter 2, I report on my research in P4 programmable dataplane verification. The focus lies on zero-specification verification of undefined behaviors in P4 dataplanes. Section 2.5 describes a new algorithm to infer runtime filters or produce code fixes in such way as to guarantee no bugs are present at runtime. Section 2.6 offers a detailed design and implementation report of P4 verification tools.

In chapter 3, I dive into network dataplane equivalence, another implicit form of specifying network correctness. The observation is that virtual networks are often regarded from distinct perspectives – what the tenant had intended and what the provider has actually implemented. The gap between these perspectives is bridged by a highly complex piece of distributed middleware which is prone to bugs and inconsistencies. The aim of this paper is to show correctness by checking equivalence between the two perspectives. Using this method, we have come across rare bugs in Neutron – OpenStack's networking driver.

Chapter 4 presents my contributions to network control-plane simulation in large cloud networks. In section 4.1, I focus on the methods to make BGP simulation in DC networks scale to large production DCs.

Chapter 5 draws the conclusions of this dissertation and relates them to the research questions initially formulated. It also provides insights into research avenues which I find interesting to explore in the future.

Table of Contents