Formal models of software-defined networks

Feng, H.

Summary

SDN (Software-Defined Networking) represents a revolutionary approach to network architecture that enables the dynamic and flexible management of network resources through software-based control. It achieves this by decoupling the control plane from the data plane and leveraging software platforms for programmable control. The control plane lies controller, which manages, controls, and formulates policies for the entire network. It collects network state information, develops forwarding policies, and instructs the underlying network devices through southbound interfaces (e.g., OpenFlow) on how to forward data packets. The data plane consists of various network devices (e.g., switches) that are responsible for forwarding data packets based on instructions from the control plane. This thesis introduces the idea of SDN and OpenFlow protocol in Chapter 2, then presents the formal expressions in the following chapters.

Chapter 3 delves into the Reo coordination language and its application in modeling SDNs. Reo is presented as a powerful tool for specifying and orchestrating the behavior of reactive and distributed systems through the use of connectors, which are graph-based representations of data flows and synchronization constraints. Furthermore, it also discusses the composition of constraint automata, which enables the modular construction of complex systems by combining simpler components. Several basic Reo connectors are introduced, and their corresponding constraint automata are described, demonstrating how these models can be used to represent and analyze various networking scenarios. The chapter concludes by illustrating how Reo and constraint automata provide a rigorous and flexible framework for modeling the intricate behaviors of SDNs, laying the groundwork for the formal verification and analysis of these systems in subsequent chapters.

The research in Chapter 4 presents a novel Reo-based model that accurately represents the behavior of SDN switches and controllers. This model also abstracts the OpenFlow communication protocol, providing a formal framework for the implementation and analysis of SDNs.

Chapter 5 uses Promela to implement our constraint automata model of SDN, and verify the formal model by SPIN model checker. To enable rigorous verification of SDN

properties, the Reo models are translated into Promela which is the input language for the SPIN model checker. This translation facilitates the formal verification of essential SDN characteristics, including reachability, consistency, and the correctness of network policies.

Moreover, the exploration in Chapter 6 extends NetKAT, a formal language for specifying network policies, to support concurrency through the introduction of ports. This extension enhances NetKAT's capability to model stateful and concurrent systems, aligning it with the Reo-based formalism and broadening its applicability to more complex SDN scenarios.

To pioneer the integration of causality reasoning into SDN models, the approach in Chapter 7 is crucial for diagnosing network anomalies and preventing hazardous events, laying the groundwork for more sophisticated causality analyses in future work.