A Declarative Framework for the Verification of Network Protocols

Jiefei Ma, Alessandra Russo
Imperial College London
{*j.ma,a.russo*}*@imperial.ac.uk*

Jorge Lobo
ICREA - Universitat Pompeu Fabra
jorge.lobo@upf.edu

Franck Le
IBM Watson Laboratory, US
fle@us.ibm.com

The verification of network protocols is a challenging problem. Traditional model checking requires a translation into an intermediate language and suffers from state explosion. Recent proposals [1, 2, 4] have aimed at mitigating these issues. In this poster [1], we propose a different and novel declarative framework to tackle this problem. The framework builds upon the recent realization that with simple extensions, database-style query languages (e.g., Datalog) can be used to specify and implement network protocols [3]. Contrary to imperative languages which describe *how* computation should be executed, declarative languages strive to specify *what* computation should be performed. As such, declarative languages have a clear correspondence to mathematical logic, and are well-suited for analysis. The large body of work and techniques developed by the formal logic community can be directly applied and taken advantage of to reason about network protocols.

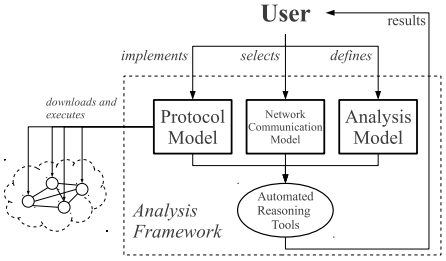

Figure 1: Analysis Framework

Our framework consists of three components (Figure 1): A *protocol model* specifies the protocol and defines the actions of a node. This protocol model is used for both analysis and direct execution. So no translation into an intermediate language for analysis is needed. The second component is a *communication model* which defines the properties of the communication links. This component is independent of the protocol specification and different communication models (e.g., *synchronous or asynchronous*) can be adopted for the analysis. The last component is the *analysis model*. It specifies the network configuration and the (protocol-specific) properties to check (e.g., convergence, loop-free routing, disjoint path routing). For the analysis task, several logic-based methods could be used. We have adopted an Answer Set Programming (ASP) solver, since ASP solvers have been shown to be very effective in solving problems of high computational complexity (e.g., NP-hard).

To demonstrate the applicability and generality of our framework, we implemented three routing protocols and verified three very different types of properties. First, the framework has revealed the presence of persistent *forwarding loops* in a network running a link state protocol. Secondly, it has discovered flaws in a MANET protocol [6] that was designed for finding disjoint paths. Finally, it has verified convergence in BGP with dispute wheels that are twice larger in size than those considered by prior solutions, with example execution traces upon violation and without false positive answers.

Thanks to its declarative nature, our declarative implementations are more concise and permit a considerably more compact representation of nodes' internal states, allowing our framework to scale the analysis to larger networks than what prior approaches have enabled [4, 5].

[1] Poster submission only.

References

[1] GUERRAOUI, R., AND YABANDEH, M. Model checking a networked system without the network. In *NSDI* (2011).

[2] KILLIAN, C., ANDERSON, J. W., JHALA, R., AND VAHDAT, A. Life, death, and the critical transition: finding liveness bugs in systems code. In *NSDI* (2007).

[3] LOO, B. T., CONDIE, T., GAROFALAKIS, M., GAY, D. E., HELLERSTEIN, J. M., MANIATIS, P., RAMAKRISHNAN, R., ROSCOE, T., AND STOICA, I. Declarative networking. *CACM* (2009).

[4] MUSUVATHI, M., AND ENGLER, D. R. Model checking large network protocol implementations. In *NSDI* (2004).

[5] WANG, A., TALCOTT, C., GURNEY, A., LOO, B., AND SCEDROV, A. Reduction-based formal analysis of bgp instances. *TACAS* (2012).

[6] ZHANG, Y., WANG, G., HU, Q., LI, Z., AND TIAN, J. Design and performance study of a topology-hiding multipath routing protocol for mobile ad hoc networks. In *IEEE INFOCOM* (2012).