Phoenix: A System for Automatically Reconfiguring Networks

Theophilus Benson, Aditya Akella
University of Wisconsin, Madison

David Maltz
Microsoft Research

1. MOTIVATION

Modern enterprise networks have complex topologies and impose a variety of restrictions on network communication. Enterprise networks also evolve with time due to software patches, worm outbreaks, network upgrades etc. These characteristics of enterprises generally tend to make their **network configuration files** highly complex.

The high degree of configuration complexity makes enterprise networks error-prone, difficult to troubleshoot and difficult to modify. When problems arise or when the network's configuration needs to change, operators are often forced to intervene manually. Since their time is at a premium, operators generally implement quick hacks, which further exacerbates the complexity. In particular, the quick-fixes make configurations of different devices just slightly different from each other (and therefore difficult to track), this makes network-wide configuration incomprehensible, and even small changes become very difficult to make over time.

There is anecdotal evidence that network operators do perform periodic readjustments of their networks to aid future management tasks. However, the techniques they employ today are manual, ad hoc and time-consuming. As a result, network-wide readjustments are undertaken only when the situation is too dire, or the readjustments are made to wait until a large network infrastructure upgrade happens. Also, the current techniques are error-prone because operators cannot easily reason about the correctness and completeness of the readjustment they have made: The readjusted configuration may not faithfully reproduce the network-wide reachability constraints from before the change.

Thus, we argue that it is important to develop a framework to systematically and automatically reorganize the existing configuration of a network into a simpler, equivalent form. Armed with such a framework, operators can get away with making local spot changes and periodically invoking the framework to re-adjust their network to a simpler but equivalent state. Thus, such a framework would make sure that the network configuration remains as simple as possible even in the face of arbitrary manual quick-fix changes, In the long run, this makes the network less prone to errors and requires less operator intervention. In fact, the simplified network facilitates quick and accurate spot-changes.

2. CHALLENGES

We propose a candidate framework called Phoenix for simplifying the configuration of a network. Phoenix performs this simplification by searching through the space of all equivalent configuration files for configuration files with the least amount of complexity. Implementing Phoenix entails several challenges:

- **Equivalent Configurations:** Phoenix must determine the set of all configuration that are equivalent to the current configuration of the network. We define two sets of configuration as being "equivalent" if both sets produce the same reachability profiles [2].
- **Evaluating Configurations:** Once the search space is defined, Phoenix must compare two sets of configurations and determine which is simpler. To address this challenge, we utilize a suite of configuration complexity metrics [1] to evaluate equivalent configurations and pick the simplest.
- **Constraints Specification:** As shown by prior work [1], some amount of complexity is necessary for financial and security reasons. Operators should be able to specify, as input into Phoenix, what portions of the configuration files are invariant to change. We are currently interviewing operators to determine a list of the most common set of constraints that operators are likely place on their network.
- **Operator Documentation:** Once a network has been reconfigured, Phoenix should generate documentation to aide operators in understanding the new network. We propose a configuration dictionary, which maps stanzas from the old configuration files to their new homes in the new configuration, i.e., what got moved and to where.

3. REFERENCES

[1] T. Benson, A. Akella, and D. A. Maltz. Unraveling the complexity of network management. In *NSDI*, 2009.
[2] G. Xie, J. Zhan, D. A. Maltz, H. Zhang, A. Greenberg, G. Hjalmtysson, and J. Rexford. On static reachability analysis of IP networks. In *Proc. IEEE INFOCOM*, 2005.