Summary

Efficient Network Reachability Analysis Using a Succinct Control Plane Representation

Seyed K. Fayaz and Tushar Sharma, Carnegie Mellon University; Ari Fogel, Intentionet; Ratul Mahajan, Microsoft Research; Todd Millstein, University of California, Los Angeles; Vyas Sekar, Carnegie Mellon University; George Varghese, University of California, Los Angeles

Keywords:

- network availability, network reachability analysis
- control plane, data plane
- Data plane analysis, Control plane analysis
- diverse protocols (e.g., BGP, OSPF route advertisements), BDD

Motivation:

- Violations of intended reachability policies (e.g., “Can A talk to B?”) can compromise availability, security, and performance of the network
- What is critically missing today is the ability to efficiently find network reachability bugs across multiple possible environments
- The paper motivates reasoning about multiple network incarnations using real reachability bugs encountered in a large cloud provider’s network. These bugs were latent and manifested only under certain environments.
    - Maintenance triggered
    - Announcement triggered
    - Failure triggered

Related Work:

- Data plane analysis: Verifying the reachability behavior of the data plane has been widely studied. Data plane verification and testing is fundamentally limited, as a network is in a constant churn.
- Control plane analysis: Moving from the data plane to the control plane potentially enables more powerful analysis, as the former is generated by the latter.
- Clean-slate control plane design: Metarouting, glue logic, and Propane aim to build a correct by-design control plane.

ERA Overview:

- We can think of the data plane as a function of the form DP : (pkt, port) → (pkt, port). The data plane itself is generated by the control plane function given routers' configuration files, the network topology (i.e., which router ports are inter-connected), and the current environment.
- We say data plane DP is policy-compliant if $\varphi_{A \to B}$ (pkt, DP) evaluates to true for all A-to-B packets.
- Rather than producing the data plane that results from a given environment, we can analyze the control plane under that environment to determine:
    - the routes that each router in the network learns via its neighbors
    - the best route when multiple routes to the same prefix are learned
- Challenges:
    - An expressive and tractable control plane model
    - Scalable control plane exploration

Modeling the Control Plane:

- To strike a balance between expressiveness and tractability, we introduce the notion of an abstract route as a succinct yet expressive I/O unit for the control plane model.
- The fields of our route abstraction are:
    - Destination IP and mask
    - Administrative Distance
    - Protocol attributes
- We lift our router model to work simultaneously on a set of route announcements. We refer to our router model as the visibility function because it captures how the router control plane processes the routing information made visible (i.e., given as input) to it.
- For fast exploration, we use BDDs to symbolically encode the set of I/O routes in a router model.

Exploring the Model:

- We leverage our BDD-based encoding to devise a set of scalable exploration mechanisms that use (i) the Karnaugh map, (ii) equivalence classes, and (iii) vectorized CPU instructions.
- Conceptually, each iteration of ERA over a BDD input analyzes a set of concrete environments for which the network has an identical behavior.

Going beyond reachability:

- Building on basic A-to-B reachability, ERA can be used to check a wider range of policies.
    - Valley-free routing
    - Blackhole-freeness
    - Waypointing
    - Loop-freeness

Implementation:

- We implement the control plane model, the K-map, and atomic predicates in Java.
- A natural question might be how much effort it takes to add support for various routing protocols to ERA. In our experience, this effort is minimal. It took two of the authors a few hours to model the common routing protocols because of two reasons.
- We evaluate ERA and find that:
    - It can help find both known and new reachability violations
    - It can scale to large networks (e.g., it can analyze a network with over 1,600 routes in 6 seconds), and our design choices are key to its scalability

Check your understanding:

- Are there any particular advantages of using BDDs for representing the control plane? Would there be differences if other representations were used?
- Are the advantages from using ERA over existing tools?
- Can we couple this approach with a complete redesign with regard to the future?
- What steps are involved in computing traffic reachable from A to B?