CSE 550: Systems for all

Au 2022

Ratul Mahajan
“At least 41% of all calls that attempted to use T-Mobile’s network during the outage failed, including at least 23,621 failed calls to 911.”

“[An old woman] who has dementia, could not reach [her son] after her car would not start and her roadside-assistance provider could not call her to clarify her location; she was stranded for seven hours”
Anatomy of the outage (illustration)
Anatomy of the outage (illustration)
Anatomy of the outage (illustration)

What if T-Mobile could guarantee that no traffic will transit Denver?

What if T-Mobile could predict the impact of link failure?
Network verification

 Guarantee network behavior*†

* Some aspect of behavior
† Under some assumptions
A horizontal slice of the problem

- Hardware
- Software (OS, protocols)
- Configuration, state
- Trust
- Verify

VERIFIED
The space of network verification tools

Shortest-path or policy routing? Are packet transformed? Stateless or stateful forwarding? ...

Features
- Data plane verification
- Control plane verification

# states analyzed
- One (live) state
- Some states
- All states
Data plane verification

Who can talk to whom using which *packets* and *paths* in *one state* of the network?
Can A talk to D and using which packets?
DPV idea: Ternary simulation

Union packet sets along possible paths
Solve using custom data structure or BDDs
Control plane verification

Who can talk to whom using which packets and paths in many states of the network?

Finds bugs proactively

Enables what if analysis
Verifying distributed control planes

Routers generate and process messages per low-level directives

```
OSPF INTERFACE INT2_1 METRIC 1
OSPF INTERFACE INT2_1 METRIC 1
OSPF REDISTRIBUTED CONNECTED METRIC 10
IP PREFIX-LIST PL1 DENY 192.168.0.0/16 LE 32
IP PREFIX-LIST PL1 ALLOW
ROUTE-MAP FROMR2 10
MATCH IP ADDRESS PREFIX-LIST PL1
SET LOCAL-PREFERENCE 120
```

Goal

Reason about states that emerge when many such programs run concurrently
CPV idea #1: Simulate the control plane

1. Simulate the control plane to generate data plane states
2. Use DPV to analyze the states

Can analyze *any* data plane but not *all* data planes?
CPV idea #2: Encode the fixed point

1. Valid network states are fixed points of the control plane
2. Fixed points can be formally encoded

ARC [2016] use a graph encoding (not general)
Minesweeper [2017] uses SMT encoding
Over to Aleksei and Yuan-Mao
So, what did we learn this quarter?