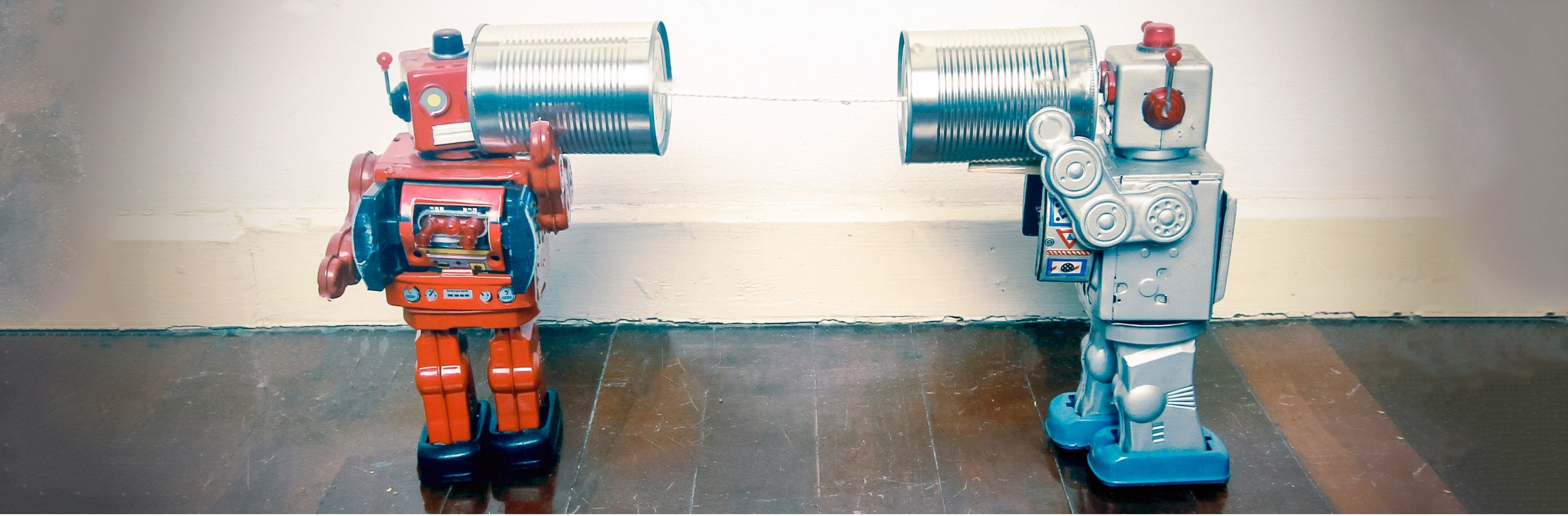


Have Your Robots Call My Robots

Using SMT Solvers for robust network automation

Simon Chatterjee

Distinguished Engineer

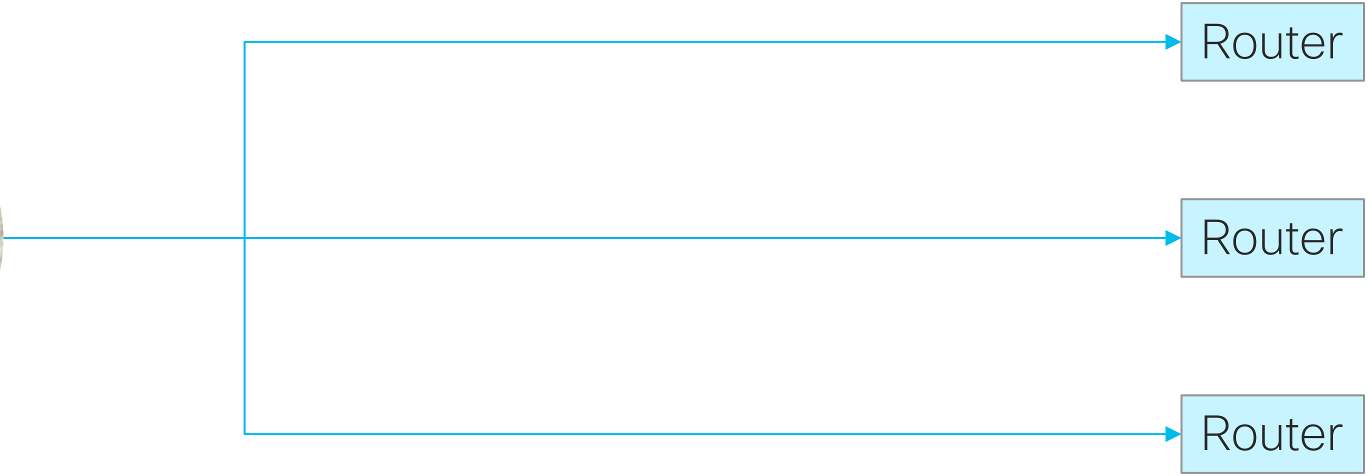
The network operations problem

- “The leading cause of Internet outages is maintenance”
- “80% of unscheduled outages are caused by people or process errors”
- “Most service providers spend 5-7x more on opex than capex”

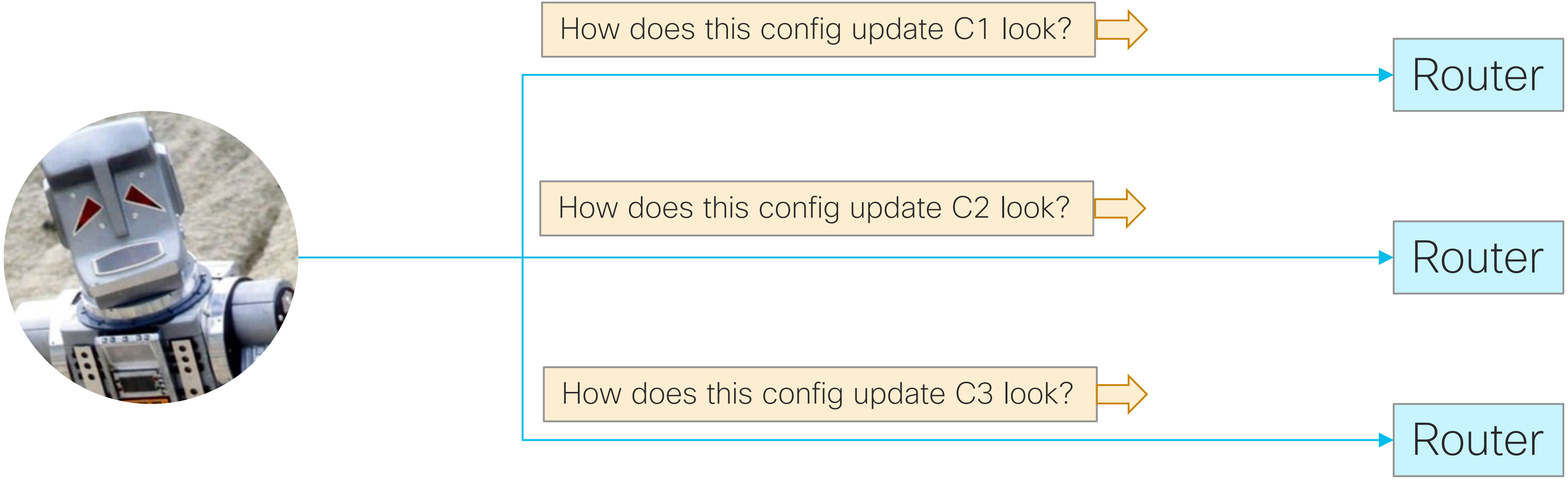
- Labovitz & Ahuja, Experimental Study of Internet Stability and Wide-Area Backbone Failures, 1999
- Scott, “Making Smart Investments To Reduce Unplanned Downtime”, 1999

The solution

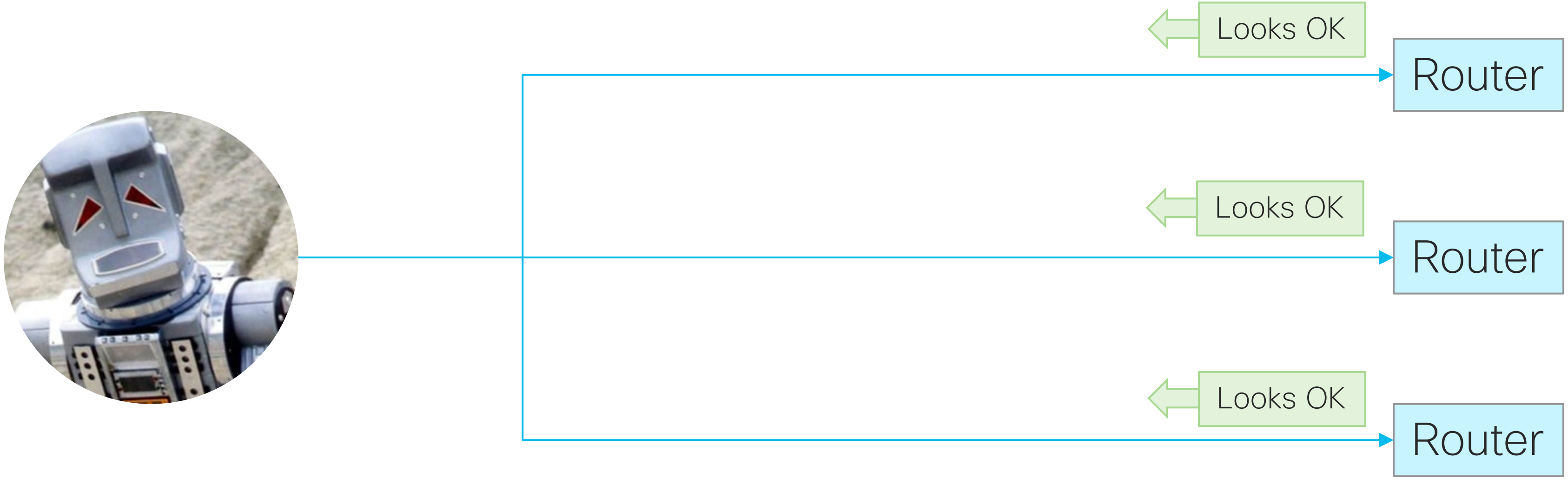
Transactionality



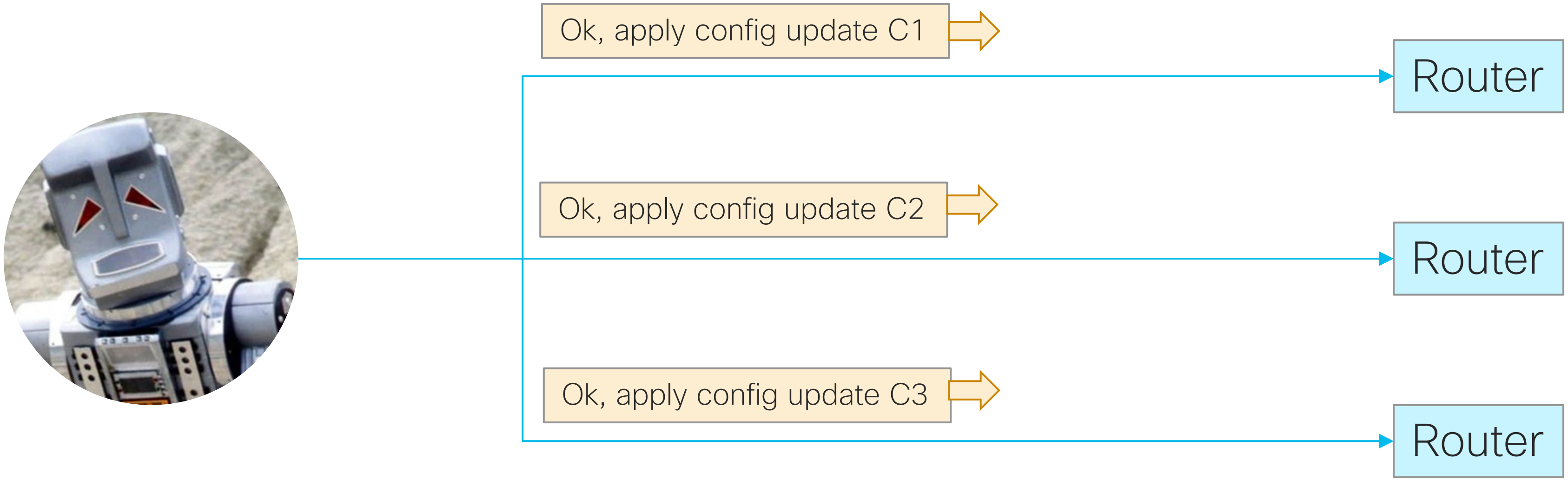
Transactionality



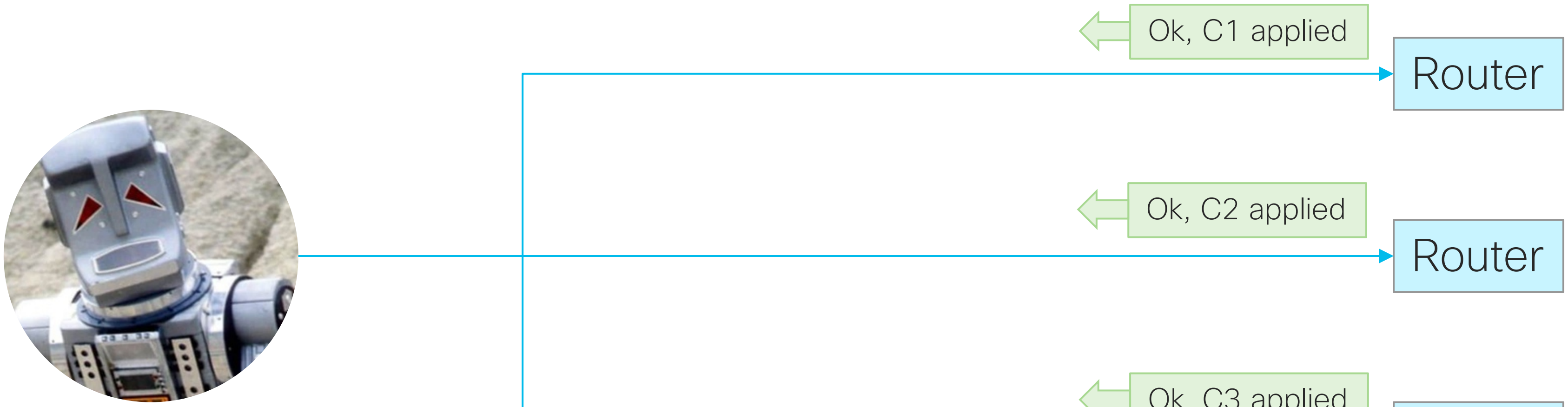
Transactionality



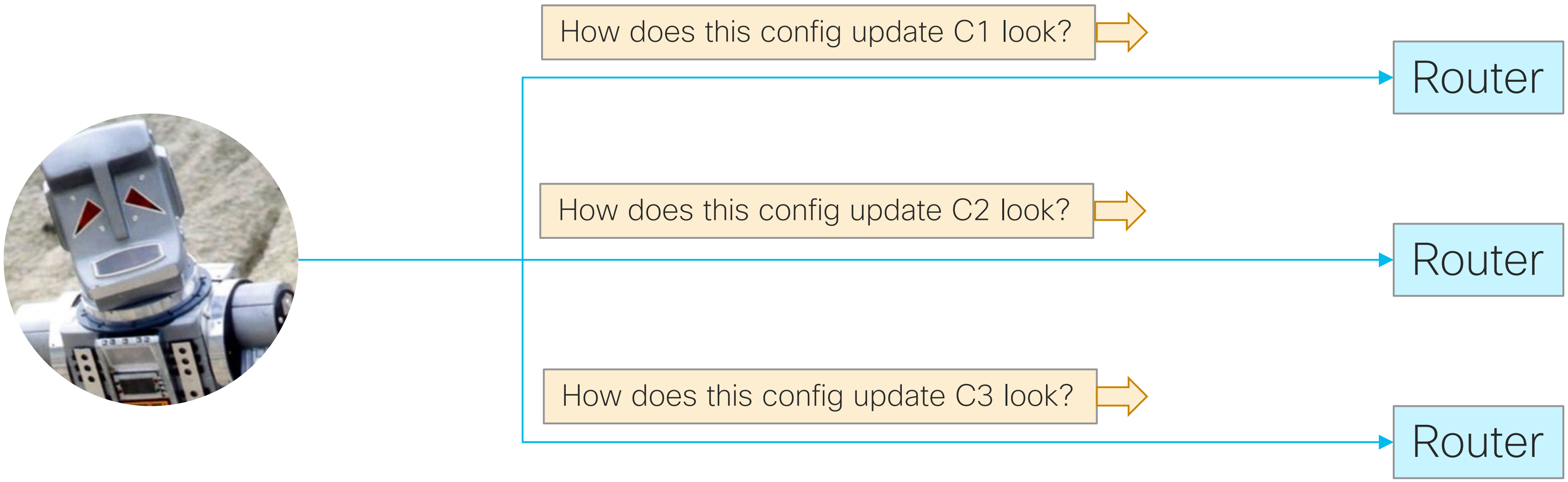
Transactionality



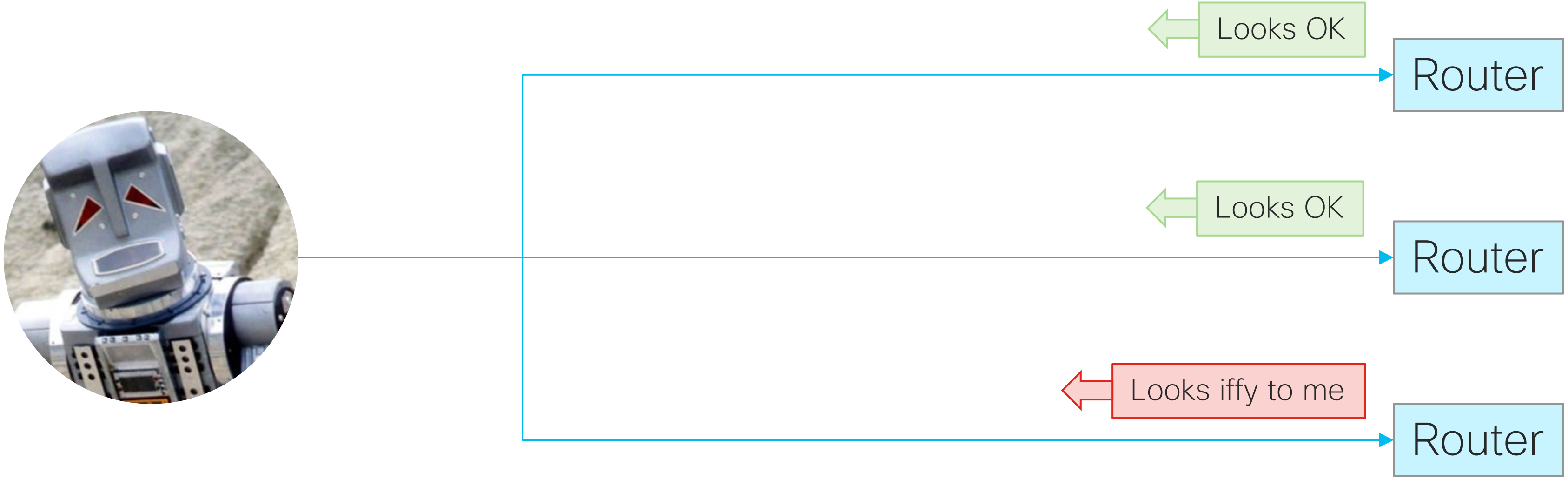
Transactionality



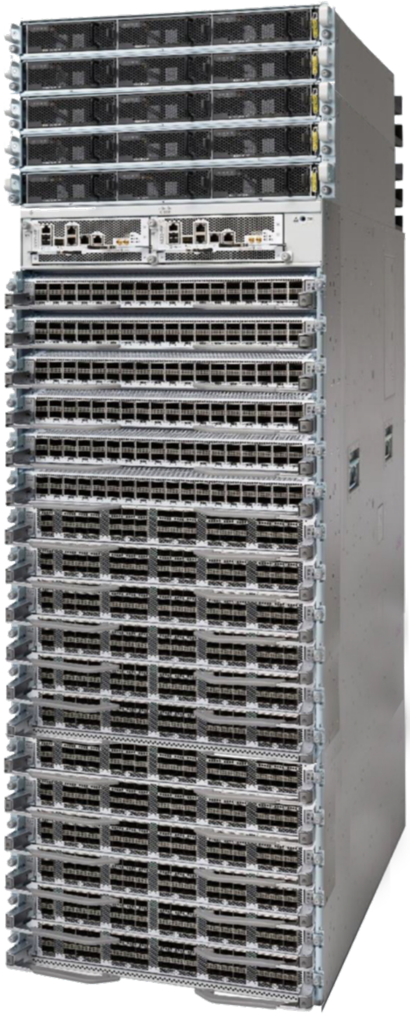
Transactionality



Transactionality



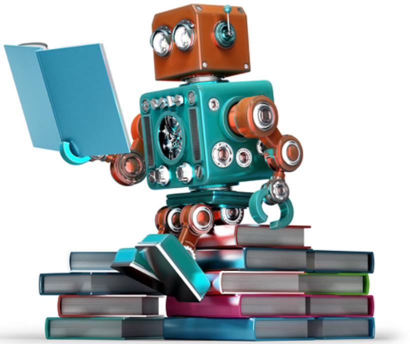
Implementing transactionality



What sort of configurations are invalid?

- VLAN tags under the same port must be unique
- There can’t be more than 64k VLAN subinterfaces on that line card type, or 256k across the whole system
- The BGP hold time must be at least double the keepalive time
- The BGP confederation AS must be different to the local AS
- Comparing BGP MED between confederation peers is valid only if the router is in a confederation (ie has a confederation AS)

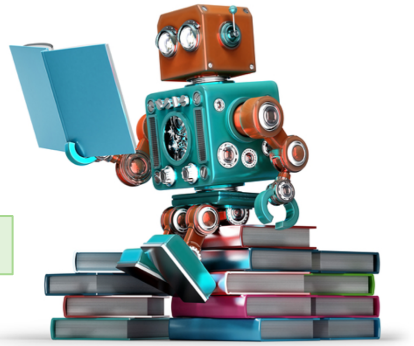
Transactionality



How does this config update C1 look?

Router

Transactionality

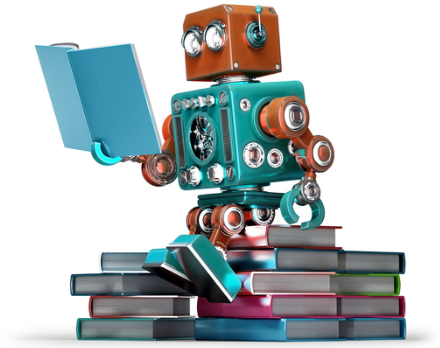


Looks OK

How does this config update C1 look?

Router

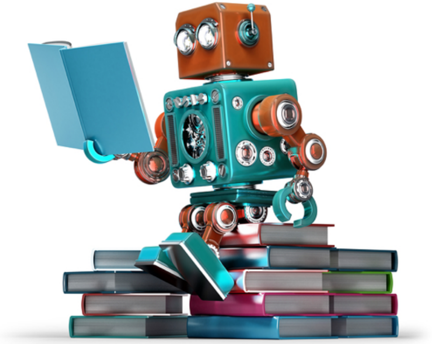
Transactionality



Okie dokie, apply config update C1 then

Router

Transactionality



Okie dokie, apply config update C1 then

Router

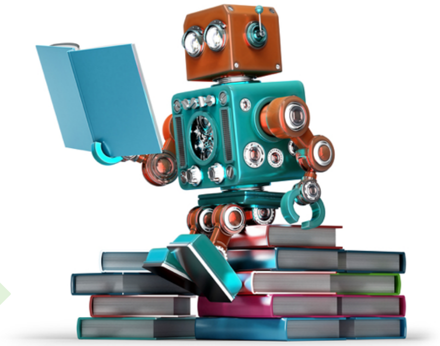
Phew! It all worked after all

Retro-fitting transactionality: ingredients

- Yang (RFC 7950): schema modeling language
  - Specifies the space of valid configuration requests
- YRL (Yang Rules Language): pure functional language to express constraints within Yang-modeled data
- YVE (Yang Validation Engine): high-performance on-box rules engine to enforce YRL-expressed constraints

Problem: rule correctness

Hand-written rules
Documenting specified behaviour



Router

Actual empirical behaviour
Implemented by millions of lines of code

Abstract problem

Program 1
(Happens to be particularly simple)

Program 2
(Happens to be particularly complex)

Do programs 1 & 2 compute the same result for every possible input?

Abstract problem

Program 1
(Happens to be particularly simple)

Program 2
(Happens to be particularly complex)

Can we automatically generate test inputs that probe every corner case of Program 1?

Example: `only-if` vs `if-and-only-if`

- `foo only-if bar` vs `foo if-and-only-if bar`

foo	bar	foo only-if bar	foo if-and-only-if bar
False	False	True	True
False	True	True	False
True	False	False	False
True	True	True	True

Example: `only-if` vs `if-and-only-if`

~~• foo~~ only-if ~~bar~~ vs foo if-and-only-if bar

`x == y`   `y > 2 * z`

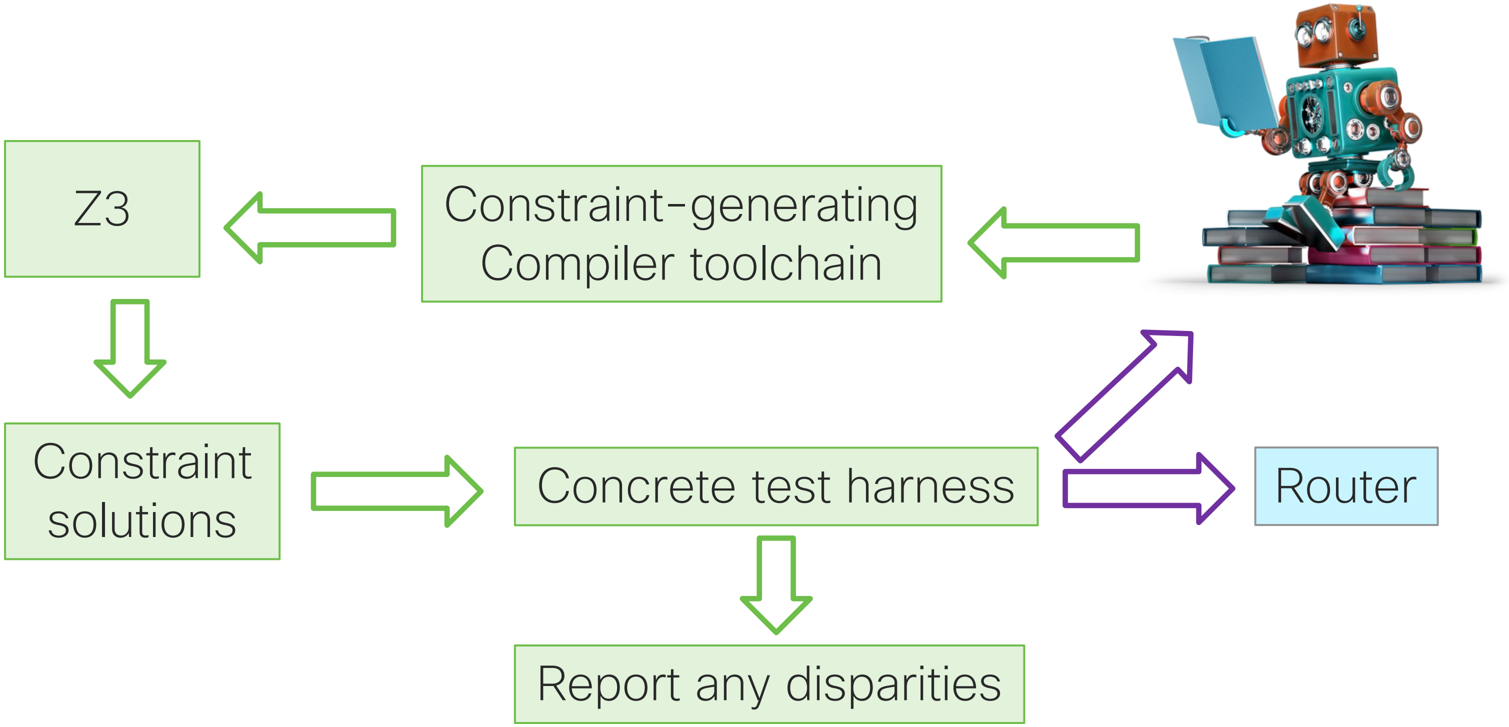
x	y	z	foo only-if bar	foo if-and-only-if bar
3	6	6	True	True
6	9	3	True	False
6	6	6	False	False
9	9	3	True	True

SMT Solvers

- Satisfiability Modulo Theories
- Extends Boolean Satisfiability solvers with select first-order theories
  - eg 64-bit two's complement bitfields, IEEE floats, finite–length arrays
- Z3 is industrial-strength, with ergonomic language bindings

- For our purposes: magic in a box
- Need just a few workarounds
  - eg for variable-sized lists, calculate a sensible finite bound and use that
  - eg for IP addresses/netmasks, use sensible templates, not random ints

Putting it together



Result

- With zero manual effort, automatically probe all* edge cases
- Common errors (such as only-if vs if-and-only-if) always* caught
- Tracks changes to code and/or rules over time automatically

* T&Cs apply, at least in theory

Stepping back

- General framework for reasoning about Yang-modeled data
  - Device-level, network-level, service-level, …
- Provides insightful developer feedback
  - eg rule X can never* fire
  - eg rule X is inconsistent* with rule Y
  - eg rule X is always subsumed by rule Y
- Formal methods in general have become vastly more approachable
  - Thank you!

* T&Cs apply, at least in theory

CISCO