Automated Network Configuration
Some Thoughts

Matthew Roughan
<matthew.roughan@adelaide.edu.au>

Discipline of Applied Mathematics
School of Mathematical Sciences
University of Adelaide

February 7, 2010
Introduction
Why do we need to take care?

I don’t know how any of the stuff works that is involved with telephone networks, let alone what works in any computer network.

former Chief Executive of a major Australian Telco
... Without Care

Pratt, Kansas

The Five Phases of Network Management
1: The Whiteboard

- Start our network management with a whiteboard picture of the network
- It’s simple, easy, and seems to work for a small network
- Until the cleaning lady cleans.
- More generally, you can’t fit a complicated network on a whiteboard.
2: Visio Diagrams

- Then we try to do the same thing, but using diagrams on a computer

- Advantages
  - Its still visual
  - We can back it up
  - Can handle more complexity than a whiteboard

- Disadvantages
  - Can’t really handle that much complexity
  - Can’t easily keep track of changes, particularly if they are being made in more than one place.
3: The Q&D Database

- Replace the diagram with a DB

- Advantages
  - Can handle as much complexity as you are prepared to store
  - Can be kept up to date

- Disadvantages
  - There is no incentive to keep the DB up to date
  - Mistakes mean DB gradually deviates from reality
4: The More Careful Database

- A Database with **Process**
- **Advantages**
  - Process forces DB to kept up to date, in theory
- **Disadvantages**
  
  In theory there is no difference between theory and practice. In practice there is.  

  *Yogi Berra*

- Process is slow (people try to circumvent it)
- Process doesn’t (usually) take into account race-like conditions
5: Realization

Realization that the database isn’t working leads to one of two approaches:

- Push
- Pull
5a: Push

Automated configuration

- Database says how the network *should* be
- Configurations are generated
- Configurations are pushed onto the routers at a regular interval (say once per day)

Limitations

- What happens if you need to change something between pushes?
- What happens when physical hardware doesn’t match expectations imposed by database?
5a: Push

Templates

Network Database

Configurations

Network
5b: Pull

- Configurations are **pulled** off the routers along with other measurements: e.g.,
  - route monitors and traceroutes
  - performance measurements
- Validation scripts test features, e.g.,
  - connectivity
- Configurations stored to form
  - database of record
  - historical data

Limitations
- Doesn’t stop problems, just finds them
- Doesn’t help scale management
5a: Pull

Network Database

Measurements

Network

Validation
The 6th Phase

Cyclic: push and pull by themselves aren't sufficient

Templates

Network Database

Configurations

Network

Measurements

Validation
The 6th Phase

More general view

- Measurement
- Decision/control
- Analysis/prediction
Features

What features are we aiming for.
Reliability

- avoid misconfigurations errors
  - these are a major source of outages
  - e.g. consistency

- two approaches
  - correctness by design (push)
  - correctness by validation (pull)

both should be used, but I don’t know which is best for most tasks.
Flexibility

- Different operators have different goals
- Don’t want unnecessarily limit
- But some limitations may be good
- e.g.,
  - C vs Java
  - Perl vs Python
- I don’t know the right level of restriction
Scale at lower cost

- Currently it is sometimes super-linear as complexity multiplies effort
  - Bottom up view of network
  - Device centric view of network
  - Irrelevant details need configuration
- Management cost should be sub-linear in number of devices
  - Top-down view
  - network-centric
  - automate irrelevant details
- What details are “irrelevant”
Commodity skills

- Current approach is device specific
  - vendor
  - model
  - OS

- Network skills should be about networks, not arcane device specific tasks
  - Analogy is move from low-level to high-level programming languages

- Maybe I want to know about a particular model of router to use a particular feature only that router has?
Plugin features

- Currently hard to add/change functionality of configuration process
  - e.g., optimization algorithm
  - e.g., new validation checks

- Should plugins be a generic architectural issue, or just one of programming?
Retain simplicity

- Whiteboard is appealing because its simple and easy
- Processes are circumvented because they are slow and painful
- How do humans interact with the system. GUIs? Text?
Robustness

To err is human,
To really stuff up you need a computer.

- rollback
- graceful failure
- How do we stop the system causing more problems than it fixes?
Modularity

- More than one person needs to work on a network at once
- Need the ability to separate non-interacting functionality, and prevent simultaneous changes to interacting functionality
- Complicated by non-locality of protocols like BGP
- Where are the boundaries in a network?
General properties

- transparency: high-level changes should produce sensible low-level changes
- insensitivity: minimal changes
- persistence: understanding of the changes needed
- BGP is already non-transparent, sensitive and non-persistent, so can our tool really achieve this?
Formal methods
Why

- high-level
- can specify network and properties unambiguously
- we can “operate” on a network
- proofs of properties
  - e.g., connectivity
  - e.g., iBGP stability
How

- abstractions
  - if we can get these right, every thing should follow
- e.g. common programming concepts
  - if/then
  - loop
  - subroutine
- network abstractions
  - I don’t know yet
  - but let’s start with the network
Conclusion

- let's go!