



Next Generation Networks: Will automation put us out of jobs?

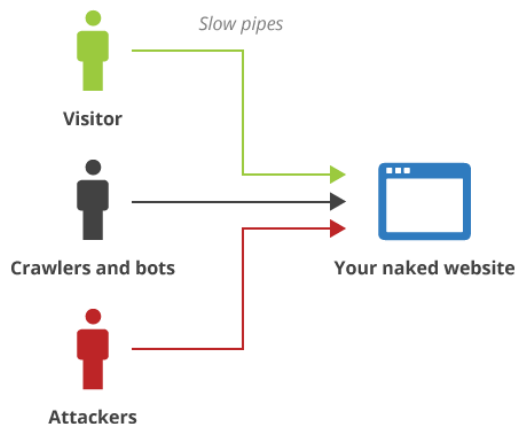
Tom Paseka
HKNOG 6.0
March 2018

About Cloudflare

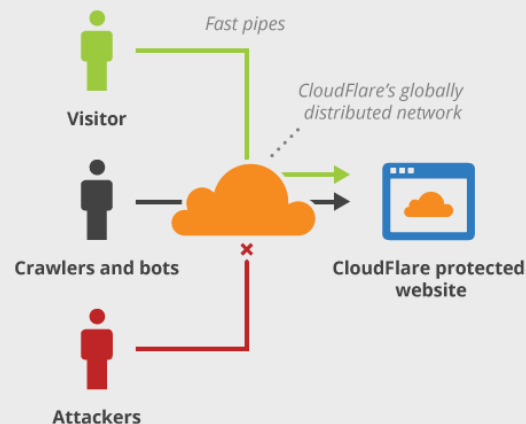
Cloudflare makes websites faster and safer using our globally distributed network to deliver essential services to any website

- Performance
- Content
- Optimisation
- Security
- 3rd party services
- Analytics

Without CloudFlare



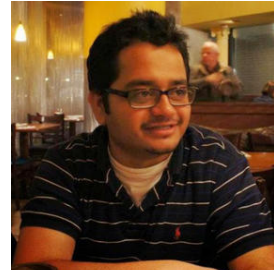
With CloudFlare



Some numbers...

- 100+ PoPs
 - 50+ countries
 - 150+ Internet exchanges
-
- >400bn Web requests a day ~10% of all web requests
 - Regular DDoS attacks larger than 500Gbps, 300M PPS

Our Team...



How?

- Supporting one of the largest global networks with a small team
- Smart, Simple design
- Automation!

Some History of Automation we've done

What we've done

- Talk I gave at APRICOT in 2015
- Basic introduction to NETCONF w/ Juniper Routers
- We used this heavily for deploying peer configuration
- But this was just a beginning.



Netconf for Peering Automation

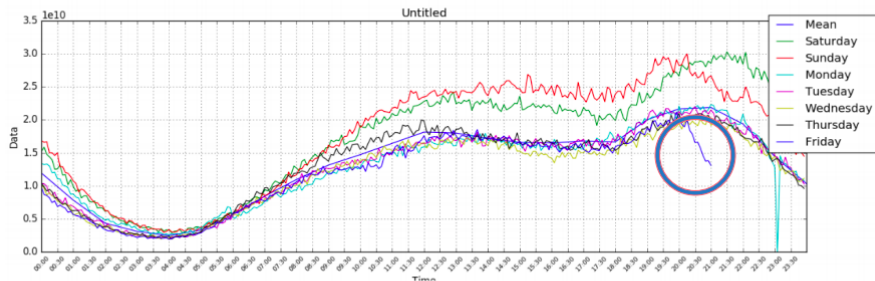
APRICOT 2015

Tom Paseka

<https://www.slideshare.net/TomPaseka/apricot-2015-netconf-for-peering-automation>

What we've done

How to monitor the Internet?



- Talk I gave in 2016
- Automation of network probing and monitoring
- Using this data and automation to remedy faults.
- Operator interaction not required for fix.

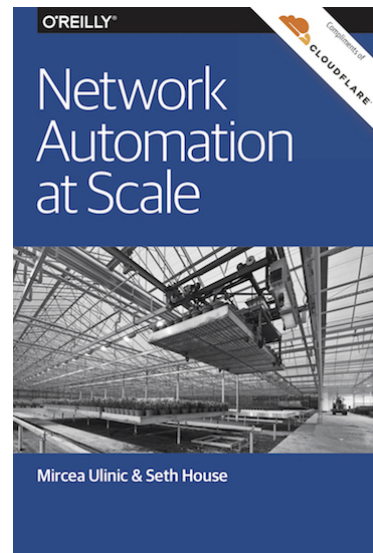
http://www.ausnog.net/sites/default/files/ausnog-2016/presentations/2.2_Tom_Paseka_AusNOG2016.pdf

What we've done

- We've open sourced much of the work we've done around automation
- Done using Salt and NAPALM
- Can grab a copy of book we wrote!

<https://www.cloudflare.com/network-automation-at-scale-ebook/>

I have a few hard copies too



Has this put us out of work?

No

What's this talk about then?

What's this talk about then?

- It's not about “SDN”, or any vendor wrapped package
- It's about work that large scale networks have been doing for a long time
- Tasks like SCPing config to a router is a perfect example
- This talk is about new networks and what we need to do, to keep up.
- This talk isn't just about right now, but 2018 and beyond.

Next Generation Networks

Next Generation Networks

- More for less
 - Larger networks, more tasks and operations
 - Less people.
- Only achievable with Automation and orchestration
- New Skills:
 - A programming language (python etc)
 - More System Administration (containers, virtualization)

Well, Has this put us out of work?

Next Generation Networks

- More “networking” work is being done by programmers or what’s typically been “systems” people.
- Network Engineers need to add more skills
- Learning a programming language will not just improve your work life, it’ll improve your worth.

Next Generation Networks

- To succeed, to compete, a change of thinking needs to happen
- Automation is the basis, not an after-thought
- Move quickly, keep moving quickly.

What does this end up looking like?

Toolings

1. salt “edge*” net.traceroute 8.8.8.8
2. salt -N EU transit.disable telia # disable Telia in EU
3. salt -G “os:junos” net.cli “show version”
4. salt -C “os:iosxr and version:6.0.2” net.arp
5. salt -G “model:MX480” probes.results
6. salt -I “type:router” ntp.set_peers 10.1.130.10 10.1.130.18
10.1.130.22

Toolings

```
edge01.sjc01:
-----
comment:
out:
-----
success:
-----
  1:
    -----
    probes:
      -----
      1:
        -----
        host_name:
          core2-1-1-0.pao.net.google.com
        ip_address:
          198.32.176.31
        rtt:
          1.507
  2:
    -----
    probes:
      -----
      1:
        -----
        host_name:
          108.170.242.225
        ip_address:
          108.170.242.225
        rtt:
          1.415
```

Toolings

```
$ sudo salt edge01.pit01 anycast.status  
edge01.pit01:
```

```
-----
```

```
anycast:
```

```
    enabled
```

```
per_plan:
```

```
-----
```

```
    biz:
```

```
        100% disabled
```

```
    ent:
```

```
        100% disabled
```

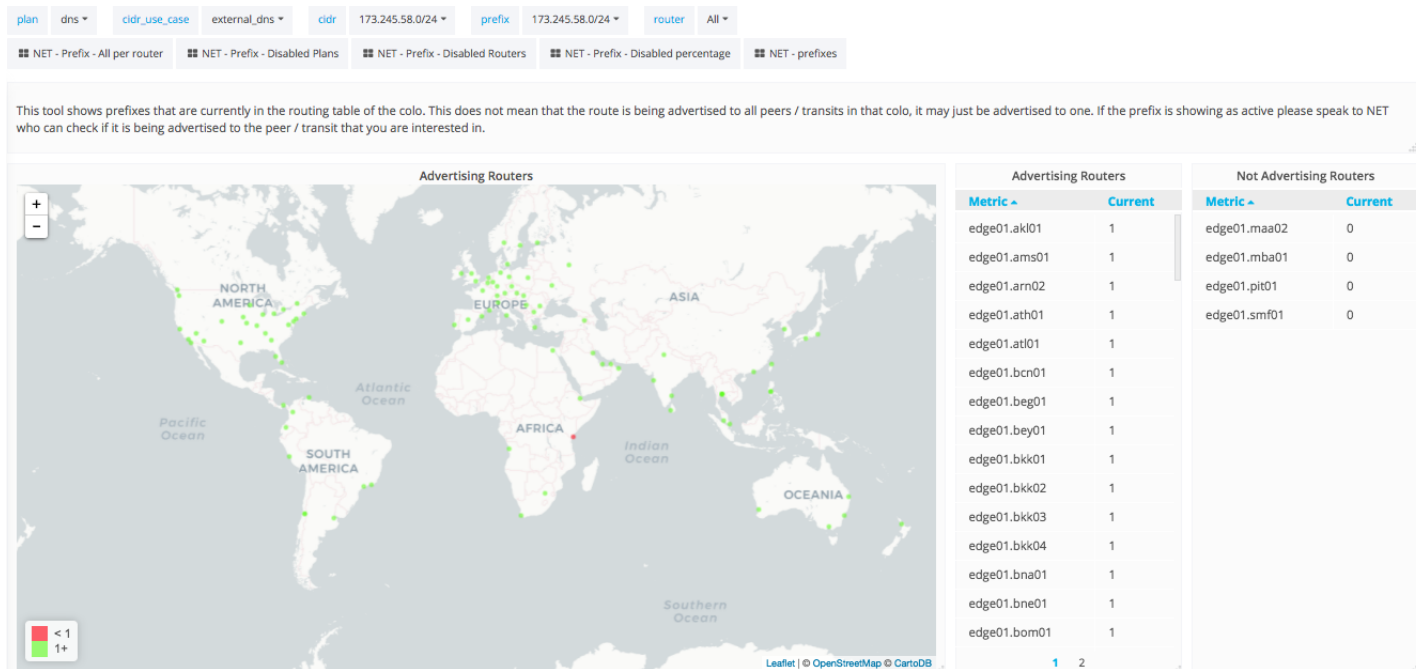
```
    free:
```

```
        42% disabled
```

```
    pro:
```

```
        100% disabled
```

Toolings



Sample YAML

```
layer2interfaces:
  - name: swp2
    type: edge
    description: "Internet Widgets Ltd"
    dot1q: yes
    shutdown: yes
    autoneg: yes
    speed: 10000
    lagindex: 1
    lagmaster: no
    fastlacp: yes
    virtualinterfaceid: 334
    vlans:
      - number: 12
        macaddress:
          - "54:1e:56:35:77:d0"
```


Sample Jinja

```
{% if pillar.get('layer2interfaces') is iterable %}
{% for iface in pillar.get('layer2interfaces') %}

default interface {{ iface.name }}
interface {{ iface.name }}
    load-interval 30

{% if iface.description|default(false) %}
    description {{ iface.description }}
{% else %}
    no description
{% endif %}

[...]

{% endfor %}
{% endif %}
```

Toolings

Use cases:

- Consistent Configuration
- Deploying rapid changes across fleet
- Reaction to changes
- Operational interface

Example

```
> show configuration services rpm | display set | match 1299 | match probe-type
set services rpm probe transit test t-edge01.scl01-1299-12956-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.eze01-1299-6762-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.lax01-1299-1299-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.eze01-1299-12956-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.mia01-1299-1299-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.lhr01-1299-1299-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.ams01-1299-1299-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.fra03-1299-1299-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.iad02-1299-1299-4 probe-type icmp-ping
set services rpm probe transit test t-edge01.sea01-1299-1299-4 probe-type icmp-ping
```

JunOS: RPM

https://www.juniper.net/documentation/en_US/junos12.1x46/topics/concept/security-rpm-overview.html

IOS-XR: ISPLA

http://www.cisco.com/c/en/us/td/docs/ios/ipsla/command/reference/sla_book/sla_02.html

Example

```
$ sudo salt-run transits.probes show_count=True  
Generated 7248 probes.
```

Generated using:

- [net.ipaddrs](#)
- [net.interfaces](#)
- [bgp.neighbors](#)
- [bgp.config](#)

All available in <https://github.com/napalm-automation/napalm-salt>

Example

```
$ cat /etc/salt/pillar/probes_edge01_dfw01.sls
```

```
probes.config:
```

```
transit:
```

```
  t-edge01.sjc01-1299-1299-4:
```

```
    source: 1.2.3.4
```

```
    target: 5.6.7.8
```

```
  t-edge01.den01-1299-1299-4:
```

```
    source: 10.11.12.13
```

```
    target: 14.15.16.17
```

```
  t-edge01.den01-174-174-4:
```

```
    source: 18.19.20.21
```

```
    target: 22.23.24.25
```

```
  t-edge01.den01-4436-4436-4:
```

```
    source: 26.27.28.29
```

```
    target: 30.31.32.33
```



```
$ sudo salt 'edge*' state.sls router.probes
```

```
edge01.dfw01:
```

```
-----
      ID: cf_probes
  Function: probes.managed
    Result: True
   Comment: Configuration updated
  Started: 23:00:17.228171
 Duration: 10.206 s
   Changes:
-----
```

```
added:
```

```
-----
transit:
```

```
-----
  t-edge01.sjc01-1299-1299-4:
```

```
-----
    probe_count:
```

```
      15
```

```
    probe_type:
```

```
      icmp-ping
```

```
    source:
```

```
      1.2.3.4
```

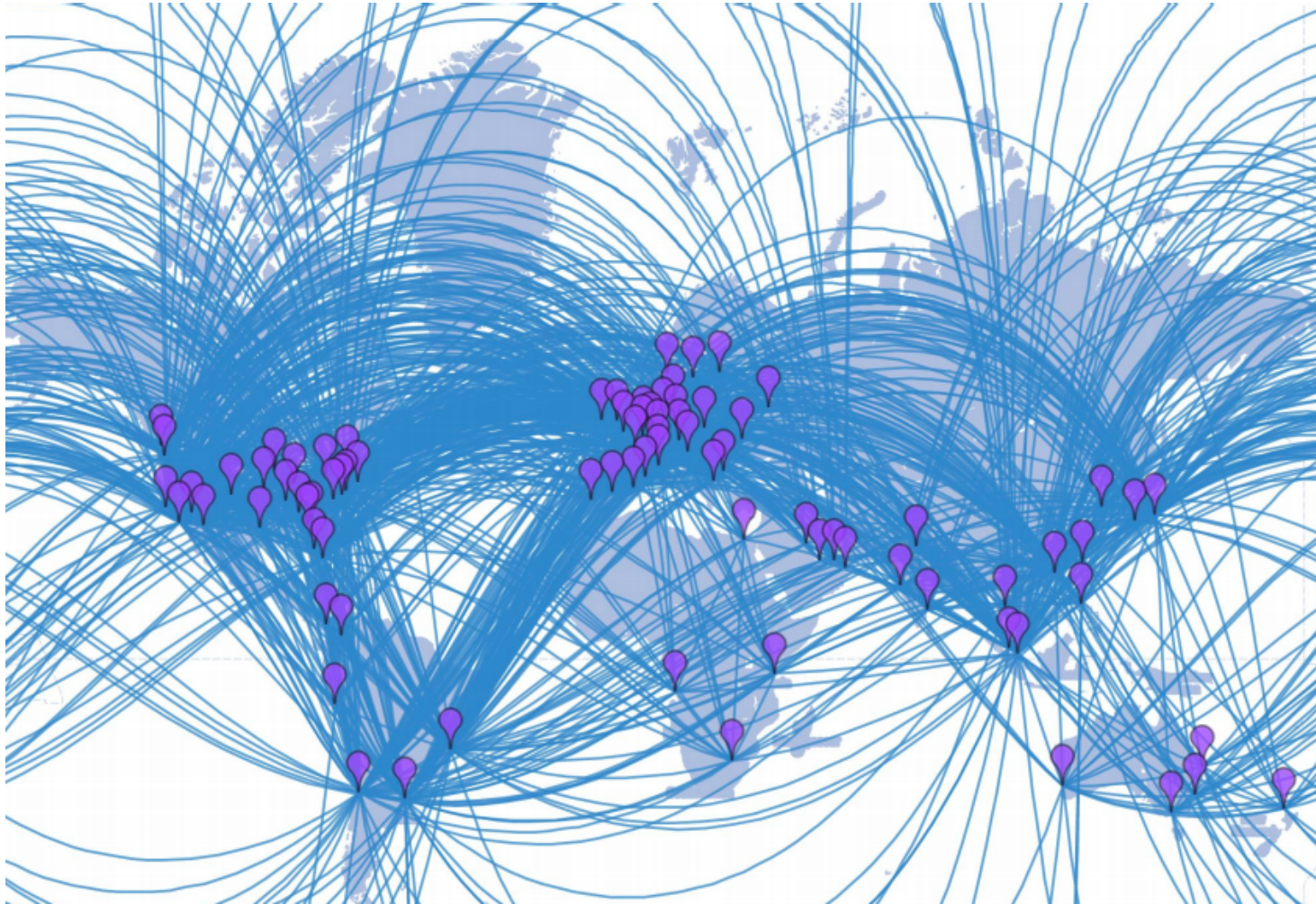
```
    target:
```

```
      5.6.7.8
```

```
    test_interval:
```

```
      3
```

Spaghetti



Example

```
$ sudo salt 'edge*' probes.results

edge01.dfw01:
-----
out:
-----
transit:
-----
  t-edge01.sjc01-1299-1299-4:
    -----
      current_test_avg_delay:
        24.023
      current_test_max_delay:
        28.141
      current_test_min_delay:
        23.278
      global_test_avg_delay:
        23.936
      global_test_max_delay:
        480.576
      global_test_min_delay:
        23.105
```

So, in summary...

Summary

- Network automation has not put us out of jobs
- New Skills are needed
- Change of thinking / methodology needed. Change your mindset!
- Better control, less errors, better networks

Questions?