Network discovery with Perl

Martin Rusko
Network Delivery Lead, HP Company

Bratislava
Perl Mongers

Agenda

- Error free network – speed/duplex audit
- Challenges we faced
- How we did it
- The procedure
- CPAN modules we used
- Final wrap up

Challenges we faced

- Network size – more than 300 router/switch pairs
- Routers owned by WAN carrier
- No discovery protocols like CDP or LLDP
- Different brands and models
- Latencies up to 500ms in APAC region

How we did it

- Key component – IP/Mac locator

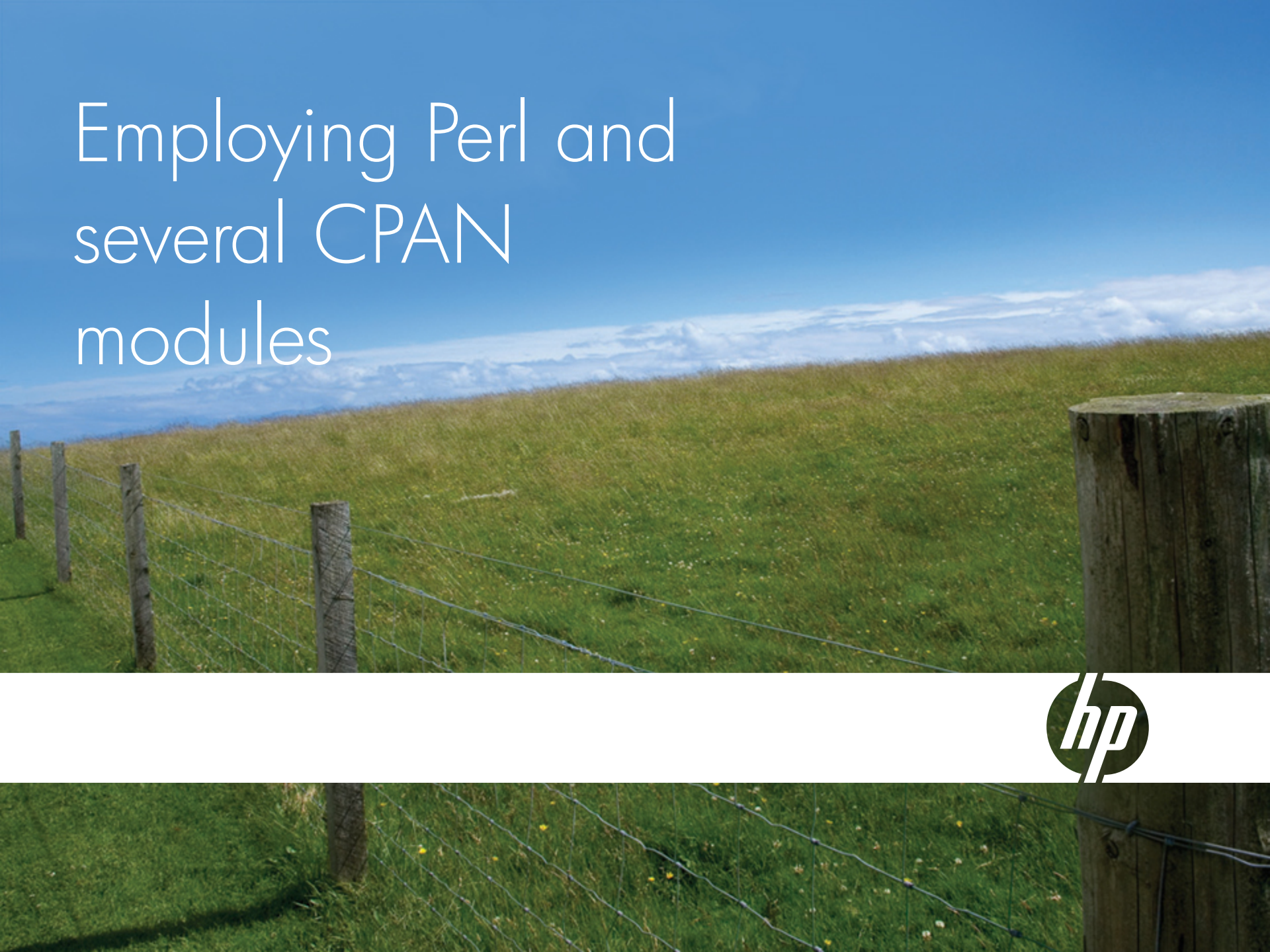
interfaces

name	ip address	type	step	connected to	on (port)	owner organization
FastEthernet0		LAN FastEthernet	pre-production			→AT&T
FastEthernet2/1/0	32.38.169.122	LAN FastEthernet	pre-production			→AT&T
FastEthernet2/1/1	32.38.57.245	LAN FastEthernet	pre-production	⇧ → kcocat1.mgt.k	→Fa3/1	→AT&T
FastEthernet2/1/2		LAN FastEthernet	pre-production			→AT&T
FastEthernet2/1/3		LAN FastEthernet	pre-production			→AT&T
GigabitEthernet0/0	10.32.29.37	LAN GigabitEthernet	pre-production	⇧ → kcocat1.mgt.k	→Gi1/48	→AT&T
GigabitEthernet0/1		LAN GigabitEthernet	pre-production			→AT&T
Loopback0	32.37.129.33	Logical Loopback	pre-production			→AT&T

The procedure

- Populate CMDB with WAN routers data
  - interface mac address
  - speed/duplex
- Let IP/MAC locator discover the links
- Poll LAN switches identified as remote endpoints
- Compare speed/duplex settings, report mismatches

Employing Perl and several CPAN modules



SNMP::Info

- module behind Netdisco tool
- device abstraction, easy to extend

```
use SNMP::Info;

my $info = new SNMP::Info( ...

# device name
my $name  = $info->name();

# CDP information
my $c_ip       = $info->c_ip();

# interface duplex
my $i_duplex   = $info->i_duplex();
```

Discovering routers

```
'siteid' => 'AU002',
        'version' => 'IOS Version 12.4(6)T7',
        'name' => 'AUBEAMSTE0002R',
        'model' => '2811',
        'hpname' => 'aubeamste0002r.mgt.',
        'interfaces' => [
                          {
                            'ip' => '138.249.122.1',
                            'name' => 'FastEthernet0/0',
                            'duplex' => 'full',
                            'r2type' => 'LAN FastEthernet',
                            'speed' => '100 Mbps',
                            'type' => 'ethernetCsmacd',
                            'address' => '138.249.122.1',
                            'mac' => '001c5829c418',
                            'iid' => '1',
                            'mask' => '255.255.255.192'
                          },
                        ],
        'address' => '10.127.252.15',
        'mac' => '001c5829c418',
        'vendor' => 'Cisco',
```

- starting with
  - site id
  - router's address
  - snmp community
- interface information
  - speed/duplex
  - mac address
- other information

Parallel::Iterator

- Provides a parallel 'map' function
- Uses forking to get multiple worker processes
- Huge performance gain for I/O intensive tasks

```
use Parallel::Iterator qw( iterate_as_array );

my @routers = iterate_as_array( { workers => 10 },
                                \&pollrouter, \@csvdata );
sub pollrouter {
        my ($array_id, $csvline) = @_;

        <-- discovering router here -->

        return $array_id, \%router;
}
```

Final wrap up

- Excellent modules available on CPAN
- Mission accomplished :-)

```
konovr1:home/hpemarus/duplex/att $ wc -l evpn.csv
311 evpn.csv
konovr1:home/hpemarus/duplex/att $ time ./get-evpn-info.pl
11346: Processing site AD001 device 10.63.252.229 ...
11347: Processing site AT001 device 10.63.252.12 ...
[...]
11347: Connected via Loopback IP 10.94.252.108
11355: Processing site ZA001 device 10.63.252.26 ...
11355: Connected via Loopback IP 10.63.252.26
11348: Processing site ZA002 device 10.63.252.27 ...
11348: Connected via Loopback IP 10.63.252.27

real     7:21.5
user       54.8
sys         2.9
```


invent