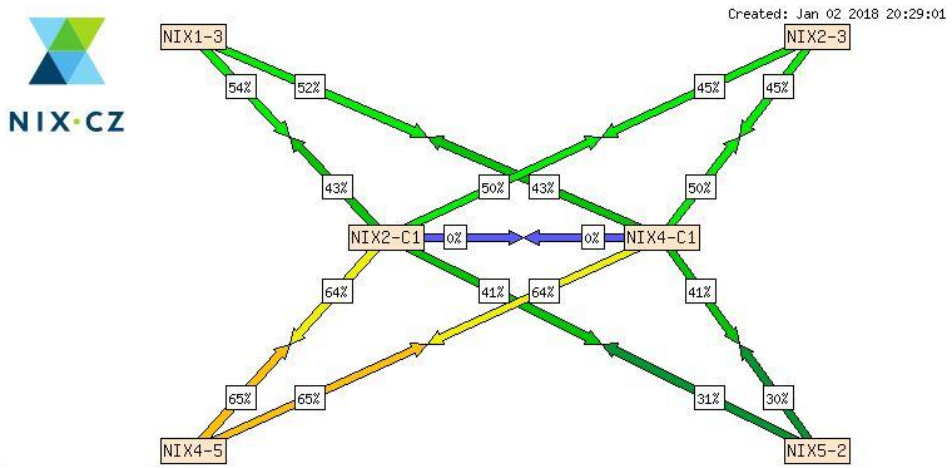
Provisioning Cisco NXOS using model driven programming

Marian Rychtecký
Radek Šenfeld

NIX·CZ

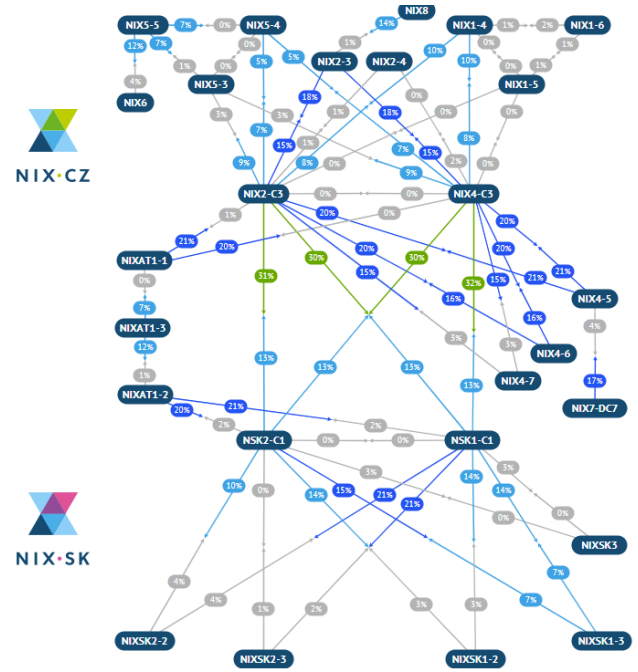
NIX.CZ motivation

- Migration from “dual-star vPC” to “leaf-spine”
- Expansion from four to nine POPs



NIX.CZ motivation

- Migration from “dual-star vPC” to “leaf-spine”
- Expansion from four to nine POPs



NIX·CZ

NIX·SK

NIX·CZ

NIX.CZ motivation

- Migration from “dual-star vPC” to “leaf-spine”
- Expansion from four to nine POPs
- Capacity upgrade (20 x 400 GE)
- IXP API

What we have tried….

“CLI based” templating with Ansible, NAPALM …

Our experience wasn’t good in terms of

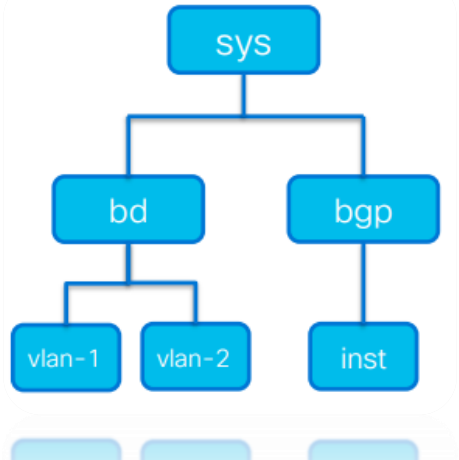
- speed
- default values issue
- reliability

What we wanted

- Fast, reliable template-based provisioning
- (streaming) telemetry

NX OS DME
(Data management engine)

- Data model available over the NX-API REST interface
- Data model is organized as a hierarchical tree made up of objects
- Represent both „all“ of the configuration features and operational data



NIX·CZ

Source https://blogs.cisco.com/developer/open-nx-os-journey-3

NX OS DME
(templates translating)

```
snmp-server contact email@domain.cz
snmp-server location Site 1, Prague, CZ
```

POST /api/mo/sys/snmp/inst.json

```
{
   "snmpInst": {
      "children": [
         {
            "snmpSysInfo": {
               "attributes": {
                  "sysContact": "email@domain.cz",
                  "sysLocation": "Site 1, Prague, CZ"
               }
            }
         }
      ]
   }
}
```

NIX·CZ

NX OS DME

Bootstrap config development took ~1 month and we are able to set

- feature sets enabling
- QoS, TACACS+ (AAA), syslog, snmp-server, NTP server, STP, CDP
- route-maps, ACLs, sFlow, ip routes
- OSPF, BGP (E-VPN), BGP culling
- Interfaces, VLANs, VNIs, port-security, loopbacks

NX OS DME

We're NOT able to set

- TCAM carving
- write (save) configuration
- „default interface ethX" (internally [un]sets many things)

NX OS DME

What's tricky

- **UDLD**
  - CLI doesn't allow you to configure without TRX
  - API does, but it's not visible in CLI
- **STP settings**

  Instead of "spanning-tree vlan 1-3967 priority 24576"

  You have to go through a loop for all VLANs

```
for vlan in range(1, 3967 + 1):
        stp.append(NexusEntity("stpVlan",
                adminSt="enabled",
                id=vlan,
                priority=stp_priority
        ))

r = nx.post(f"/api/mo/sys/stp/inst.json", payload=stp)
```

NIX·CZ

NX OS DME

Can anything go wrong? Oh, yes! ☹

```
A_NICE_SWITCH_200_MILES_AWAY# copy run start
[####################################] 100%
Configuration update aborted: request was aborted
```

NIX·CZ

NX OS DME

```
A_NICE_SWITCH_200_MILES_AWAY# copy startup-config running-config
Duplicate sequence number
Duplicate sequence number
Duplicate sequence number
Duplicate sequence number
Duplicate sequence number
Duplicate sequence number
                                        ...
```

NIX·CZ

NX OS DME

```
...

Warning: Please save config and reload the system for the configuration to take  effect
Warning: Please save config and reload the system for the configuration to take  effect
Syntax error while parsing 'switchport'
Syntax error while parsing 'switchport access vlan XYZ'
Syntax error while parsing 'mtu 9216'
Syntax error while parsing 'storm-control broadcast level 0.10'
Syntax error while parsing 'storm-control multicast level 1.00'
Syntax error while parsing 'load-interval counter 2 60'
Syntax error while parsing 'load-interval counter 3 300‘

...
```

NIX·CZ

NX OS DME

```
...

Performing image verification and compatibility check, please wait....

...
```

(Here, the switch froze for ~2 minutes - you keep praying)

NIX·CZ

NX OS DME

```
                                ...
Copy complete, now saving to disk (please wait)...
Copy complete.
```

(Finally, some relief…)

NX OS DME

Why is that?

- **We don’t know for sure**
- **We assume internally the Nexus is object-based, and CLI is emulated**
  - **Configuring objects is not translated to CLI by 100%**
  - **You can easily break things (a couple of restarts needed)**

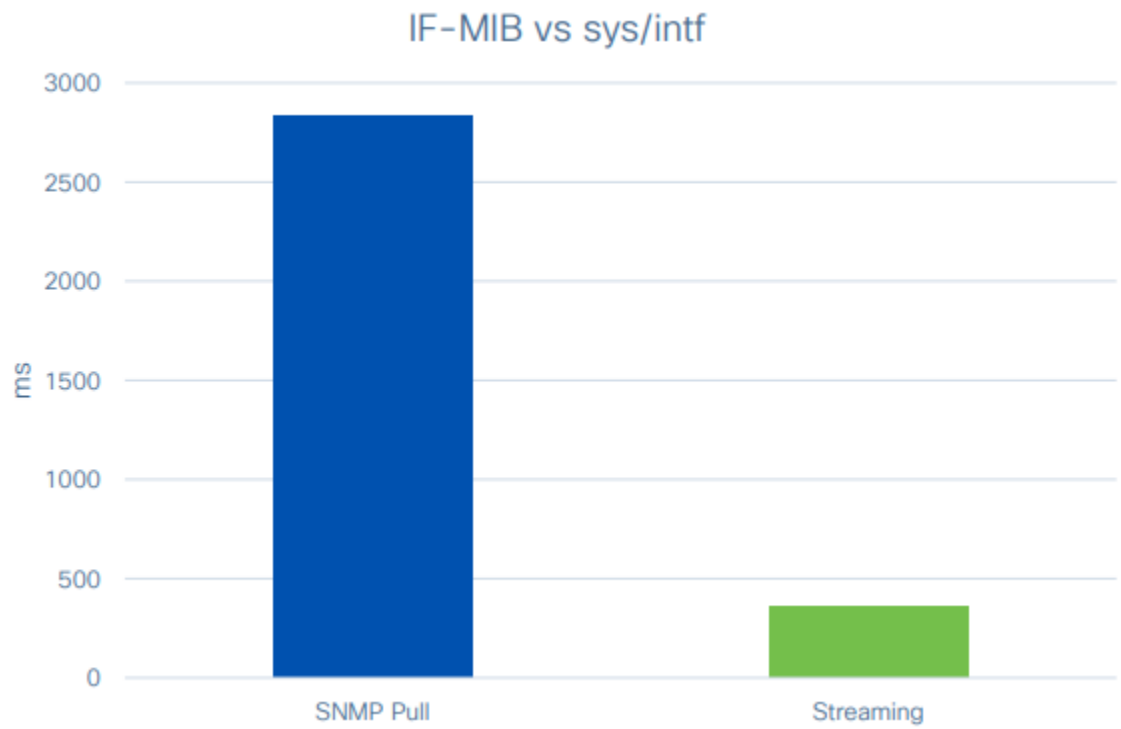
NX OS DME

What you get ?

- **Speed**
  - **Requests takes milliseconds (full switch setup ~5s)**
  - **Individual requests (interface, VLAN, VNI, BGP settings) ~100ms**
  - **Reliability of the REST API**

- **Operational parameters**

NX OS Telemetry



- SNMP bulk-get `IF-MIB`
- Streaming DME sensor path `sys/intf`

NX OS Telemetry

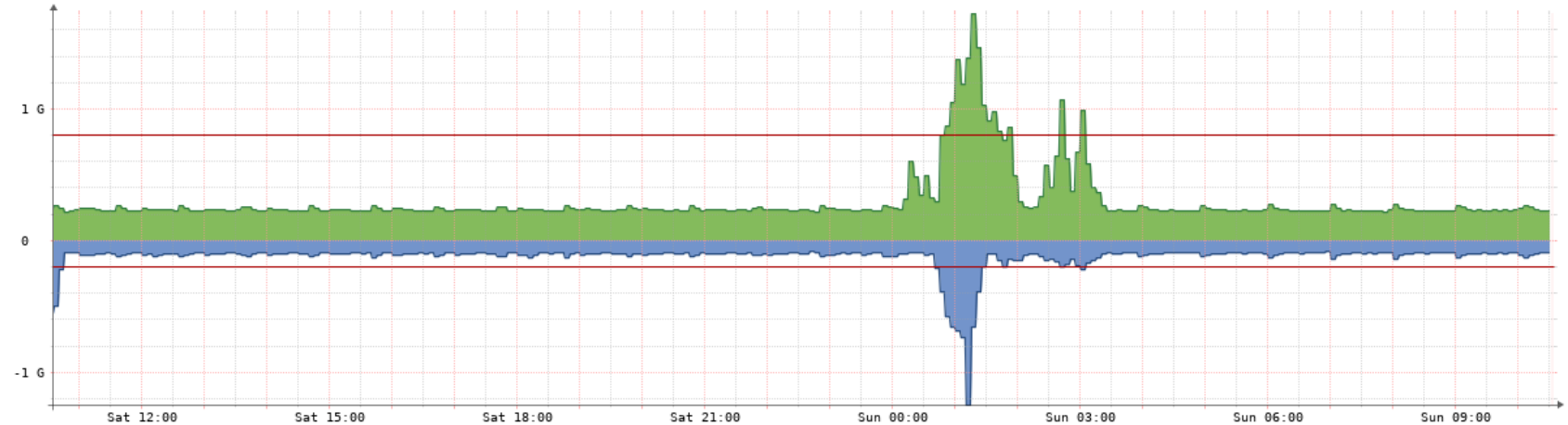
```
telemetry
  destination-group 100
    ip address 192.168.1.1 port 57000 protocol gRPC encoding GPB
    use-chunking size 4096
  sensor-group 100
    path sys/intf depth unbounded
  subscription 600
    dst-grp 100
    snsr-grp 100 sample-interval 10000
```

NX OS Telemetry

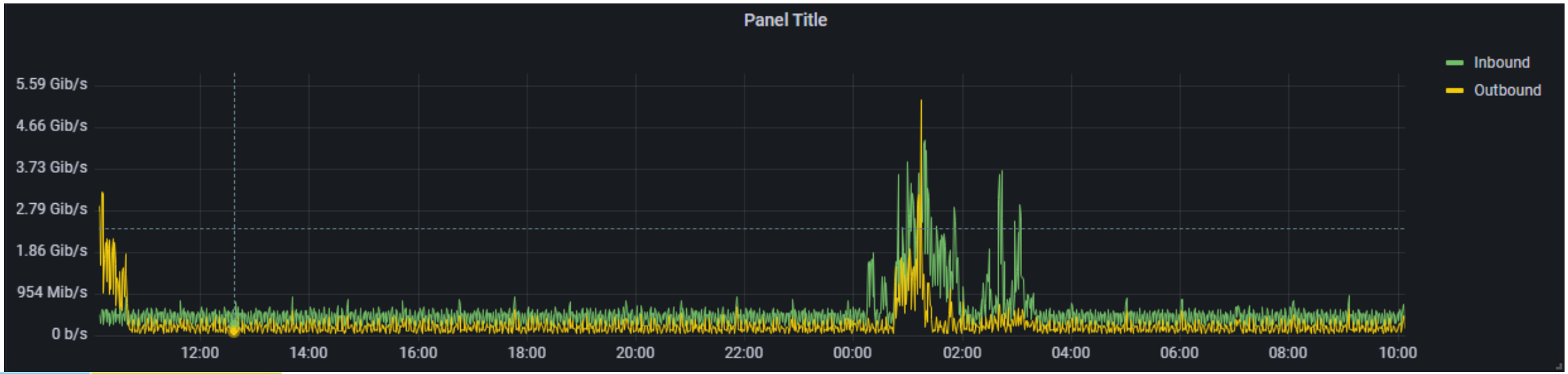
```
sys/intf(100) :
GPB Encoded Data size in bytes (Cur/Min/Max):
434528/434455/446008


Collection Time in ms (Cur/Min/Max): 375/1/538
Encoding Time in ms (Cur/Min/Max): 139/129/295
Transport Time in ms (Cur/Min/Max): 1/1/63
Streaming Time in ms (Cur/Min/Max): 525/1/753
```

NIX·CZ

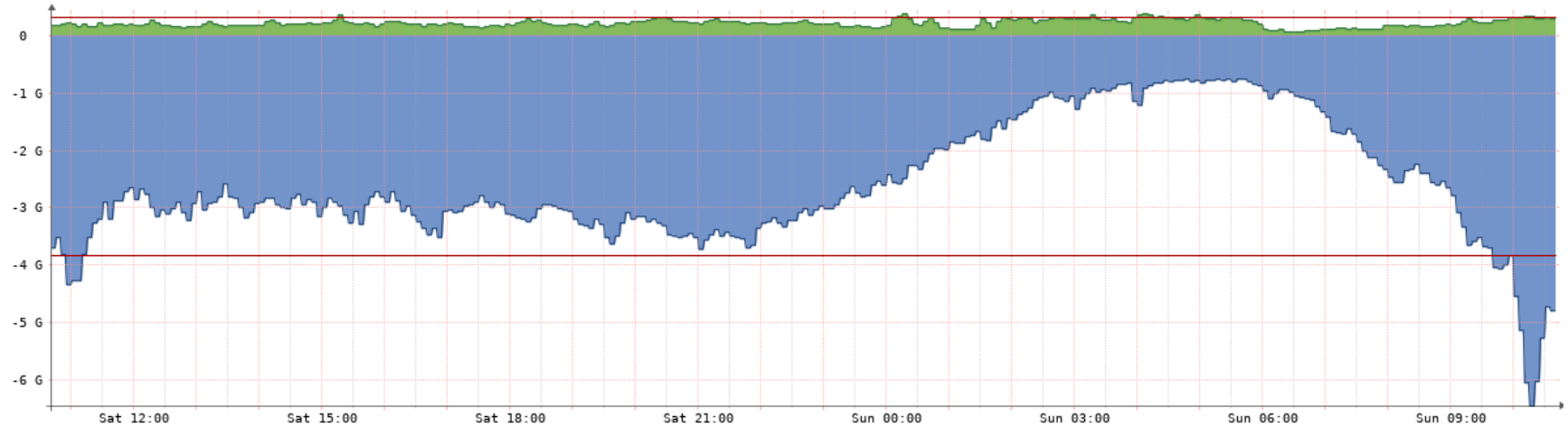


Bits/s	Last	Avg	Max	95th
In	228.63M	297.55M	1.72G	803.29M
Out	90.10M	123.55M	1.25G	198.08M
Total	4.55T (In	3.21T	Out	1.33T)



NIX·CZ

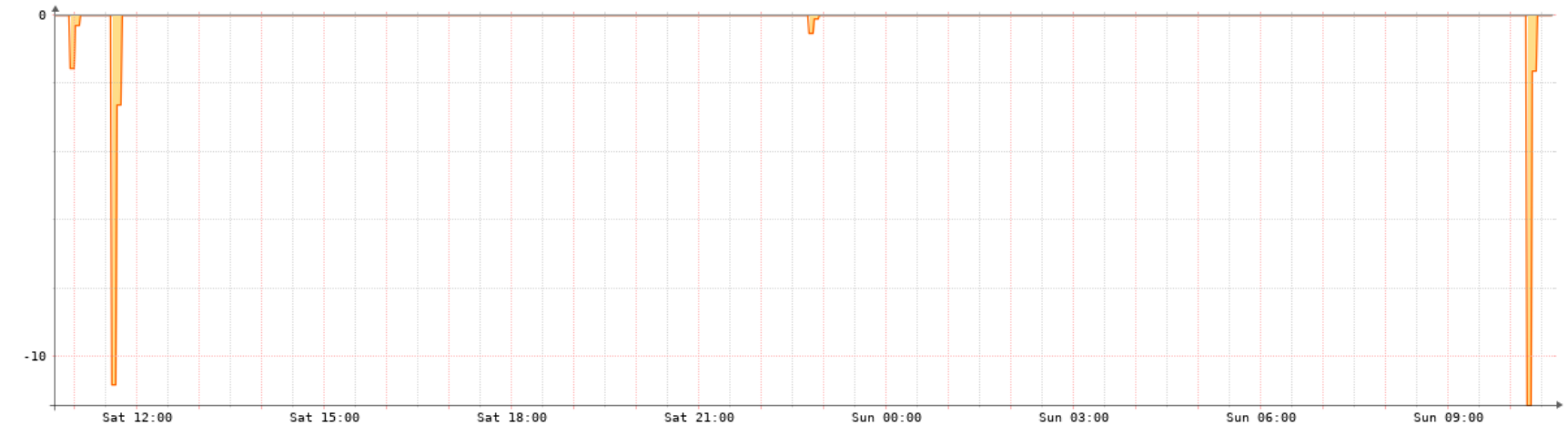
Traffic bit/s



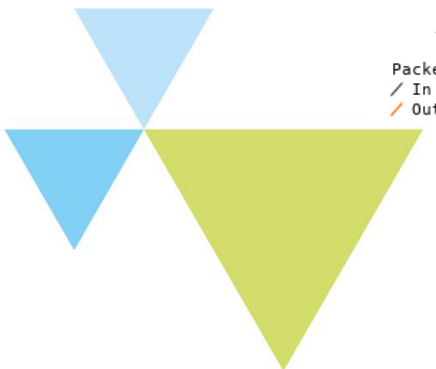
Bits/s	Last	Avg	Max	95th
In	288.46M	221.20M	388.30M	325.53M
Out	4.79G	2.59G	6.47G	3.84G

Total 30.31T (In 2.39T Out 27.92T)

Discards p/s



Packets/s	Last	Avg	Max
In	0.00	0.00	0.00
Out	0.00	101.06m	11.44



NIX·CZ

Thank you for your attention.

mr@nix.cz