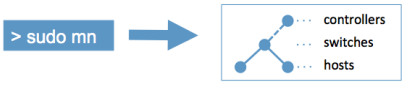
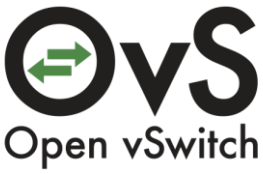
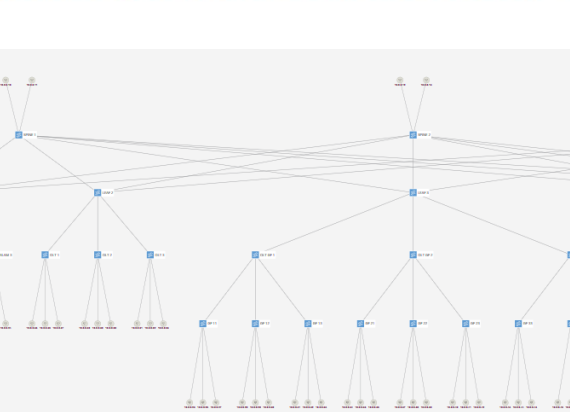
SDN Topologie WCORD







Wolfgang Hurst

1 Einleitung

1.1 Dieses Dokument und das Ziel

In diesem Dokument erstellen wir mit Mininet ein Multiples CORD Netzwerk für alle erdenklichen Experimente. Ich habe es liebevoll W-CORD genannt, das W steht für Wolfgang. Es vereint R-CORD, M-CORD, E-CORD und Services.

1.2 Voraussetzungen

Wir benötigen eine

- Mininet Umgebung => „SDN Mininet Installation“
- Mininet Topologien => „SDN Mininet Topologie“
- Laufende ONOS Umgebung => „SDN ONOS Developer Installation“

Die Mininet Umgebung sollte etwa 4 GB Ram haben.

2 Spine-Leaf Fabric

Beim SDN wird oft eine Spine-Leaf Switch Struktur verwendet. Das Konstrukt kommt dabei sowohl in Rechenzentren als auch bei CORD zum Einsatz. Es besteht aus sogenannten Spine Switches die eine Verbindung zu jedem Leaf Switch hat. Dadurch entsteht ein Block mit dem man jeden und alles auch mindestens n Wegen miteinander verbinden kann. Der ganze Block wird auch Fabric genannt.

Mit Mininet kann man solche Spine-Leaf Konstrukte ebenfalls bauen, und das wollen wir auch mal tun. Unsere Fabric soll aus drei Spine Switchen mit 6 Leaf Switchen bestehen. An jedem Leaf Switch werden diverse Services und Access Technologien angeschlossen. Alternativ was nicht dem Standard entspricht, werden wir am Spine die Backbone anbinden.

2.1 Spine Switche

Wir erstellen erst einmal drei Spine Switche

```
spine1 = self.addSwitch ("spine1", dpid="1011");
spine2 = self.addSwitch ("spine2", dpid="1012");
spine3 = self.addSwitch ("spine3", dpid="1013");
```

2.2 Leaf Switche

Wir erstellen 6 Leaf Switche

```
leaf1 = self.addSwitch ("leaf1", dpid="2021");
leaf2 = self.addSwitch ("leaf2", dpid="2022");
leaf3 = self.addSwitch ("leaf3", dpid="2023");
leaf4 = self.addSwitch ("leaf4", dpid="2024");
leaf5 = self.addSwitch ("leaf5", dpid="2025");
leaf6 = self.addSwitch ("leaf6", dpid="2026");
```

2.3 Links der Spine-Leaf

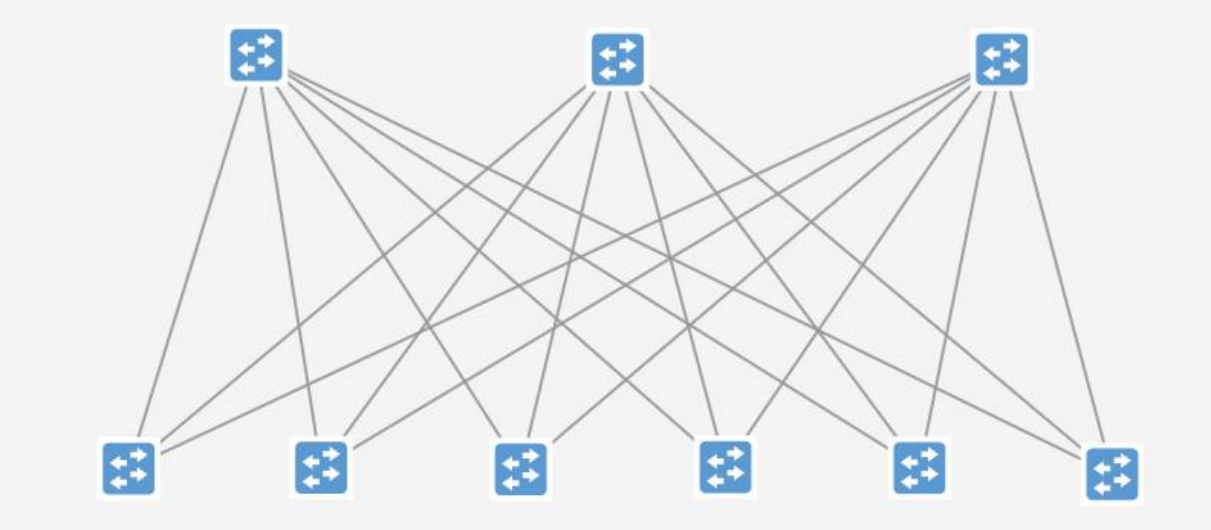
Jetzt muss man die Spine Switche mit den Leaf Switchen verbinden. Dabei gehen wir so vor das wir am Spine ab Port 11 auf die Leaf Switche auf ab Port 1 geht. Die Ports 1-10 auf dem Spine reservieren wir für die Backbone Anbindung. Durch die Versetzung weiß man das Port 1-10 immer ein „Uplink“ nach oben ist

```
# Links zwischen Spine 1 und allen Leafs
self.addLink (spine1, leaf1, 11, 1);
self.addLink (spine1, leaf2, 12, 1);
self.addLink (spine1, leaf3, 13, 1);
self.addLink (spine1, leaf4, 14, 1);
self.addLink (spine1, leaf5, 15, 1);
self.addLink (spine1, leaf6, 16, 1);

# Links zwischen Spine 2 und allen Leafs
self.addLink (spine2, leaf1, 11, 2);
self.addLink (spine2, leaf2, 12, 2);
self.addLink (spine2, leaf3, 13, 2);
self.addLink (spine2, leaf4, 14, 2);
self.addLink (spine2, leaf5, 15, 2);
self.addLink (spine2, leaf6, 16, 2);

# Links zwischen Spine 3 und allen Leafs
self.addLink (spine3, leaf1, 11, 3);
self.addLink (spine3, leaf2, 12, 3);
self.addLink (spine3, leaf3, 13, 3);
self.addLink (spine3, leaf4, 14, 3);
```

```
                self.addLink (spine3, leaf5, 15, 3);
                self.addLink (spine3, leaf6, 16, 3);
```

2.4 Vorläufiges Ergebnis

Das ganze sollte dann in etwa so aussehen



Oben die Spine und unten die Leaf

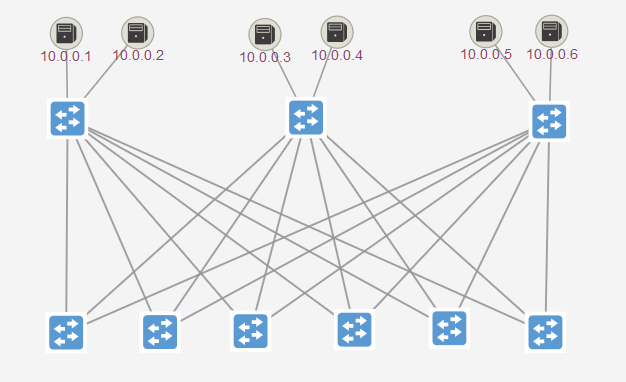
3 Spine Based Backbone

Entgegen den Empfehlungen werden wir im W-CORD die Backbone am Spine terminieren.

3.1 Core Router

Für eine Backbone Anbindung an den Spine definieren wir jeweils zwei Hosts pro Spine die als Backbone Router agieren können

```
cr11 = self.addHost ('cr11'); self.addLink (spine1, cr11, 1, 1);
cr12 = self.addHost ('cr12'); self.addLink (spine1, cr12, 2, 1);

cr21 = self.addHost ('cr21'); self.addLink (spine2, cr21, 1, 1);
cr22 = self.addHost ('cr22'); self.addLink (spine2, cr22, 2, 1);

cr31 = self.addHost ('cr31'); self.addLink (spine3, cr31, 1, 1);
cr32 = self.addHost ('cr32'); self.addLink (spine3, cr32, 2, 1);
```



4 Leaf Access Netze

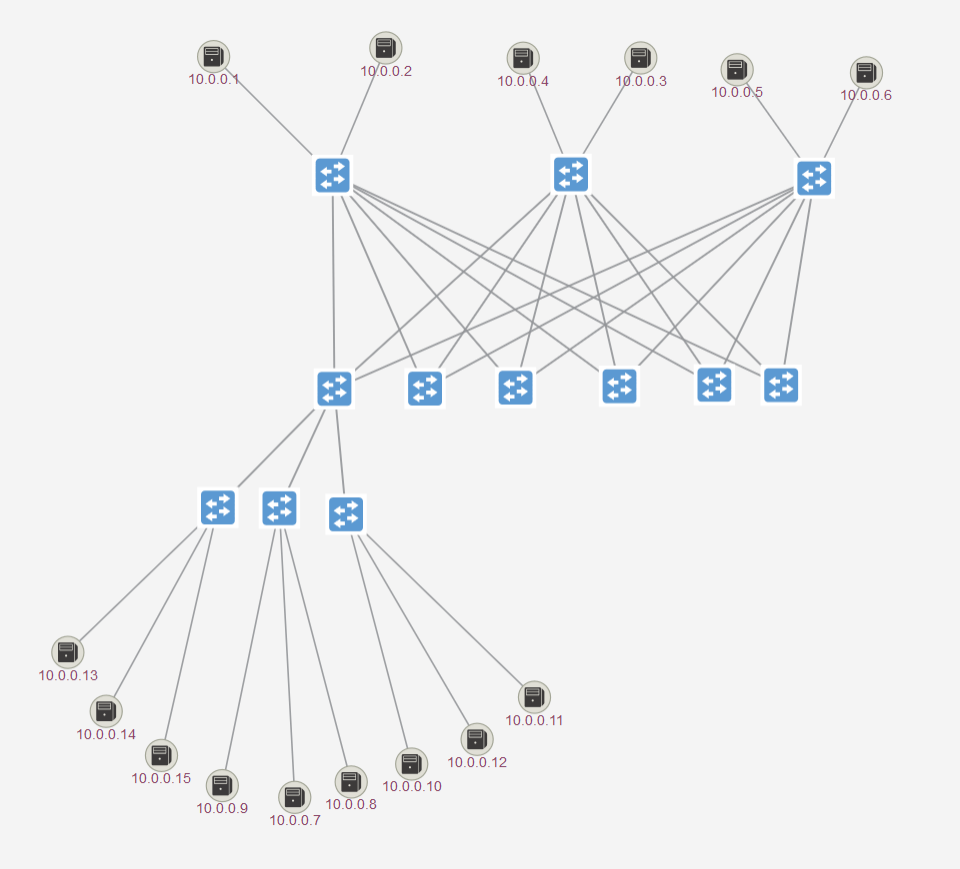
4.1 R-CORD

Wir erstellen am Leaf 1 und Leaf 2 und Leaf 3 vier Varianten von R-CORD. Einmal DSL per DSLAM, einmal OLT mit GPON G.Fast und einmal pures OLT/ONT und einmal CFV. Da wir die Geräte im Mininet so nicht ohne weiteres Simulieren können, nehmen wir dazu erst einmal Switche.

4.1.1 DSLAM

Wir erstellen drei DSLAMs mit jeweils drei CPEs dahinter und klemmen alles an Leaf 1

```
        rdslam1 = self.addSwitch ("rdslam1", dpid="21011");
        self.addLink (rdslam1, leaf1, 1, 11);

        rdslam2 = self.addSwitch ("rdslam2", dpid="21012");
        self.addLink (rdslam2, leaf1, 1, 12);

        rdslam3 = self.addSwitch ("rdslam3", dpid="21013");
        self.addLink (rdslam3, leaf1, 1, 13);

        dsl11 = self.addHost ('dsl11'); self.addLink (dsl11, rdslam1, 1, 11);
        dsl12 = self.addHost ('dsl12'); self.addLink (dsl12, rdslam1, 1, 12);
        dsl13 = self.addHost ('dsl13'); self.addLink (dsl13, rdslam1, 1, 13);

        dsl21 = self.addHost ('dsl21'); self.addLink (dsl21, rdslam2, 1, 11);
        dsl22 = self.addHost ('dsl22'); self.addLink (dsl22, rdslam2, 1, 12);
        dsl23 = self.addHost ('dsl23'); self.addLink (dsl23, rdslam2, 1, 13);

        dsl31 = self.addHost ('dsl31'); self.addLink (dsl31, rdslam3, 1, 11);
        dsl32 = self.addHost ('dsl32'); self.addLink (dsl32, rdslam3, 1, 12);
        dsl33 = self.addHost ('dsl33'); self.addLink (dsl33, rdslam3, 1, 13);
```

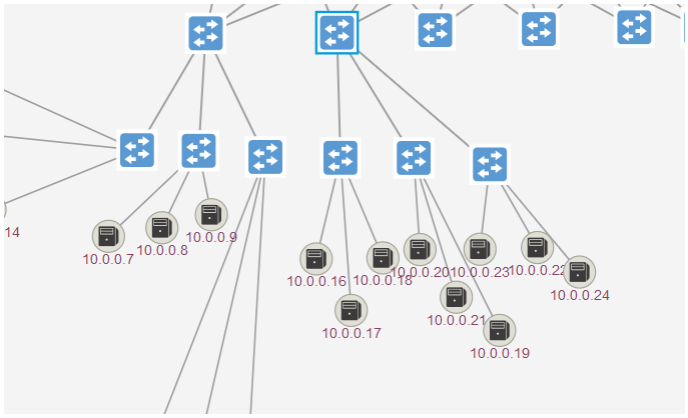


4.1.2 OLT / ONT

Ähnlich wie bei DSL erstellen wir 3 OLT mit jeweils 3 ONT am Leaf 2

```
        rolt1 = self.addSwitch ("rolt1", dpid="21021");
        self.addLink (rolt1, leaf2, 1, 11);

        rolt2 = self.addSwitch ("rolt2", dpid="21022");
        self.addLink (rolt2, leaf2, 1, 12);

        rolt3 = self.addSwitch ("rolt3", dpid="21023");
        self.addLink (rolt3, leaf2, 1, 13);

        ont11 = self.addHost ('ont11'); self.addLink (ont11, rolt1, 1, 11);
        ont12 = self.addHost ('ont12'); self.addLink (ont12, rolt1, 1, 12);
        ont13 = self.addHost ('ont13'); self.addLink (ont13, rolt1, 1, 13);

        ont21 = self.addHost ('ont21'); self.addLink (ont21, rolt2, 1, 11);
        ont22 = self.addHost ('ont22'); self.addLink (ont22, rolt2, 1, 12);
        ont23 = self.addHost ('ont23'); self.addLink (ont23, rolt2, 1, 13);

        ont31 = self.addHost ('ont31'); self.addLink (ont31, rolt3, 1, 11);
        ont32 = self.addHost ('ont32'); self.addLink (ont32, rolt3, 1, 12);
        ont33 = self.addHost ('ont33'); self.addLink (ont33, rolt3, 1, 13);
```

Und so langsam kommen wir an die Grenze des „automatischen“ Topologie Graphen …



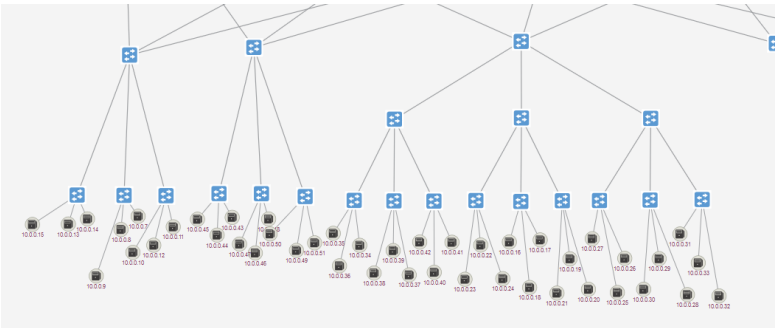
Das sieht richtig mistig aus. Und wir sind noch nicht fertig.

4.1.3 OLT / G.FAST

Dort haben wir vom OLT jeweils G.FAST „mini-DSLAMs“ im Einsatz. Wenn man sie mal so nennen darf. Wir erstellen an Leaf 3 jeweils 3 OLT mit jeweils 3 CPEs.

```
                roltgf1 = self.addSwitch ("roltgf1", dpid="21031"); self.addLink (roltgf1, leaf3, 1, 11);
                roltgf2 = self.addSwitch ("roltgf2", dpid="21032"); self.addLink (roltgf2, leaf3, 1, 12);
                roltgf3 = self.addSwitch ("roltgf3", dpid="21033"); self.addLink (roltgf3, leaf3, 1, 13);

                rgf11 = self.addSwitch ("rgf11", dpid="31011"); self.addLink (rgf11, roltgf1, 1, 11);
                rgf12 = self.addSwitch ("rgf12", dpid="31012"); self.addLink (rgf12, roltgf1, 1, 12);
                rgf13 = self.addSwitch ("rgf13", dpid="31013"); self.addLink (rgf13, roltgf1, 1, 13);

                dslgf111 = self.addHost ('dslgf111'); self.addLink (dslgf111, rgf11, 1, 11);
                dslgf112 = self.addHost ('dslgf112'); self.addLink (dslgf112, rgf11, 1, 12);
                dslgf113 = self.addHost ('dslgf113'); self.addLink (dslgf113, rgf11, 1, 13);

                dslgf121 = self.addHost ('dslgf121'); self.addLink (dslgf121, rgf12, 1, 11);
                dslgf122 = self.addHost ('dslgf122'); self.addLink (dslgf122, rgf12, 1, 12);
                dslgf123 = self.addHost ('dslgf123'); self.addLink (dslgf123, rgf12, 1, 13);

                dslgf131 = self.addHost ('dslgf131'); self.addLink (dslgf131, rgf13, 1, 11);
                dslgf132 = self.addHost ('dslgf132'); self.addLink (dslgf132, rgf13, 1, 12);
                dslgf133 = self.addHost ('dslgf133'); self.addLink (dslgf133, rgf13, 1, 13);

                rgf21 = self.addSwitch ("rgf21", dpid="31021"); self.addLink (rgf21, roltgf2, 1, 11);
                rgf22 = self.addSwitch ("rgf22", dpid="31022"); self.addLink (rgf22, roltgf2, 1, 12);
                rgf23 = self.addSwitch ("rgf23", dpid="31023"); self.addLink (rgf23, roltgf2, 1, 13);

                dslgf211 = self.addHost ('dslgf211'); self.addLink (dslgf211, rgf21, 1, 11);
                dslgf212 = self.addHost ('dslgf212'); self.addLink (dslgf212, rgf21, 1, 12);
                dslgf213 = self.addHost ('dslgf213'); self.addLink (dslgf213, rgf21, 1, 13);

                dslgf221 = self.addHost ('dslgf221'); self.addLink (dslgf221, rgf22, 1, 11);
                dslgf222 = self.addHost ('dslgf222'); self.addLink (dslgf222, rgf22, 1, 12);
                dslgf223 = self.addHost ('dslgf223'); self.addLink (dslgf223, rgf22, 1, 13);

                dslgf231 = self.addHost ('dslgf231'); self.addLink (dslgf231, rgf23, 1, 11);
                dslgf232 = self.addHost ('dslgf232'); self.addLink (dslgf232, rgf23, 1, 12);
                dslgf233 = self.addHost ('dslgf233'); self.addLink (dslgf233, rgf23, 1, 13);

                rgf31 = self.addSwitch ("rgf31", dpid="31031"); self.addLink (rgf31, roltgf3, 1, 11);
                rgf32 = self.addSwitch ("rgf32", dpid="31032"); self.addLink (rgf32, roltgf3, 1, 12);
                rgf33 = self.addSwitch ("rgf33", dpid="31033"); self.addLink (rgf33, roltgf3, 1, 13);

                dslgf311 = self.addHost ('dslgf311'); self.addLink (dslgf311, rgf31, 1, 11);
                dslgf312 = self.addHost ('dslgf312'); self.addLink (dslgf312, rgf31, 1, 12);
                dslgf313 = self.addHost ('dslgf313'); self.addLink (dslgf313, rgf31, 1, 13);

                dslgf321 = self.addHost ('dslgf321'); self.addLink (dslgf321, rgf32, 1, 11);
                dslgf322 = self.addHost ('dslgf322'); self.addLink (dslgf322, rgf32, 1, 12);
                dslgf323 = self.addHost ('dslgf323'); self.addLink (dslgf323, rgf32, 1, 13);

                dslgf331 = self.addHost ('dslgf331'); self.addLink (dslgf331, rgf33, 1, 11);
                dslgf332 = self.addHost ('dslgf332'); self.addLink (dslgf332, rgf33, 1, 12);
                dslgf333 = self.addHost ('dslgf333'); self.addLink (dslgf333, rgf33, 1, 13);
```

Es ist auch alles da. Allerding verliert man im ONOS so langsam den Überblick.



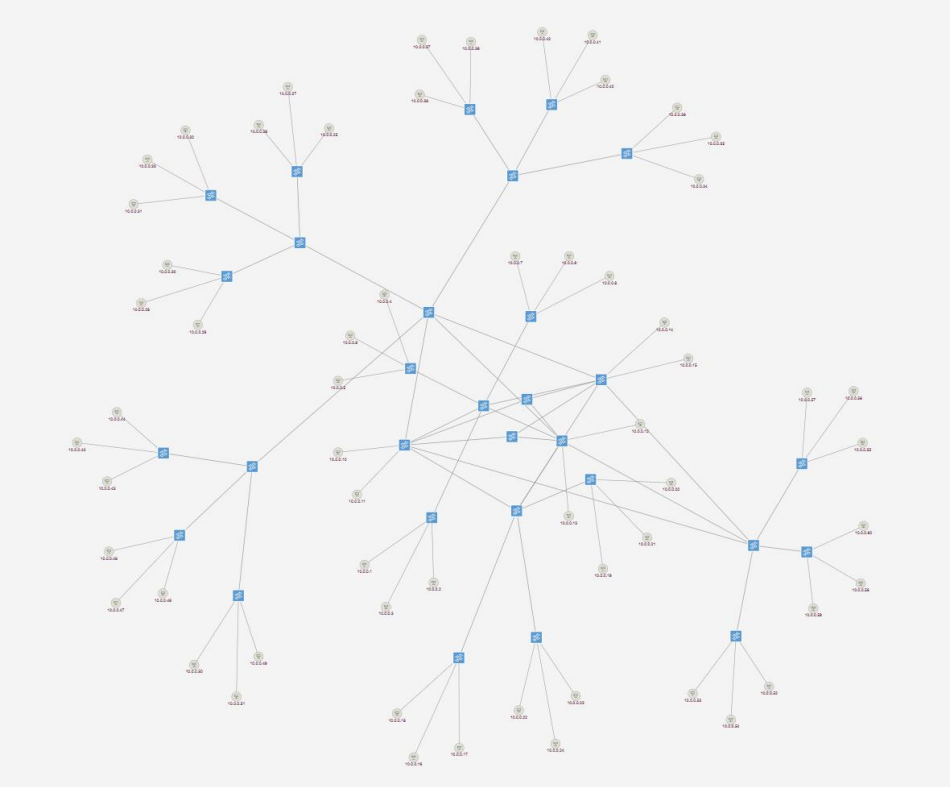
4.2 M-CORD

Am Leaf 4 hängen wir ein paar eNodeB dran und ein paar Telefone.

4.2.1 eNodeB

```
menodeb1 = self.addSwitch ("menodeb1", dpid="41011");
self.addLink (menodeb1, leaf4, 1, 11);

menodeb2 = self.addSwitch ("menodeb2", dpid="41012");
self.addLink (menodeb2, leaf4, 1, 12);

menodeb3 = self.addSwitch ("menodeb3", dpid="41013");
self.addLink (menodeb3, leaf4, 1, 13);

cp11 = self.addHost ('cp11'); self.addLink (cp11, menodeb1, 1, 11);
cp12 = self.addHost ('cp12'); self.addLink (cp12, menodeb1, 1, 12);
cp13 = self.addHost ('cp13'); self.addLink (cp13, menodeb1, 1, 13);

cp21 = self.addHost ('cp21'); self.addLink (cp21, menodeb2, 1, 11);
cp22 = self.addHost ('cp22'); self.addLink (cp22, menodeb2, 1, 12);
cp23 = self.addHost ('cp23'); self.addLink (cp23, menodeb2, 1, 13);

cp31 = self.addHost ('cp31'); self.addLink (cp31, menodeb3, 1, 11);
cp32 = self.addHost ('cp32'); self.addLink (cp32, menodeb3, 1, 12);
cp33 = self.addHost ('cp33'); self.addLink (cp33, menodeb3, 1, 13);
```

Und hier der absolut saubere und strukturierte Netzplan …



4.3 E-CORD

Am Leaf 5 legen wir für das E-CORD nur noch ein paar CPEs an einer CFV … Einfacher geht es kaum.

```
                ecpe11 = self.addHost ('ecpe11'); self.addLink (ecpe11, leaf5, 1, 11);
                ecpe12 = self.addHost ('ecpe12'); self.addLink (ecpe12, leaf5, 1, 12);
                ecpe13 = self.addHost ('ecpe13'); self.addLink (ecpe13, leaf5, 1, 13);
```

5 Service Edge

Nun haben wir noch zwei Leaf Switche frei. Dort werden wir jetzt Service Nodes konfigurieren.

Was wir brauchen sind die folgenden Service Nodes:

- BNG / BRAS
- DHCP Server (für IPoE)
- DNS Cache Server
- RADIUS/AAA Server
- Google Cache / NetFlix Multicast Distributoren

```
                sbng11 = self.addHost ('sbng11'); self.addLink (sbng11, leaf6, 1, 11);
                sbng21 = self.addHost ('sbng21'); self.addLink (sbng21, leaf6, 1, 12);

                sdhcp11 = self.addHost ('sdhcp11'); self.addLink (sdhcp11, leaf6, 1, 13);
                sdhcp21 = self.addHost ('sdhcp21'); self.addLink (sdhcp21, leaf6, 1, 14);

                sdns11 = self.addHost ('sdns11'); self.addLink (sdns11, leaf6, 1, 15);
                sdns21 = self.addHost ('sdns21'); self.addLink (sdns21, leaf6, 1, 16);

                saaa11 = self.addHost ('saaa11'); self.addLink (saaa11, leaf6, 1, 17);
                saaa21 = self.addHost ('saaa21'); self.addLink (saaa21, leaf6, 1, 18);

                scache11 = self.addHost ('scache11'); self.addLink (scache11, leaf6, 1, 19);
                scache21 = self.addHost ('scache21'); self.addLink (scache21, leaf6, 1, 20);
```

6 Zusammenfassung

Wir haben ein R-CORD mit DSL und DSLAMs am Leaf 1

Wir haben ein R-CORD mit OLT am Leaf 2

Wir haben ein R-CORD mit OLT/G.FAST am Leaf 3

Wir haben ein M-CORD mit eNodeB am Leaf 4

Wir haben ein E-CORD mit CVF am Leaf 5

Wir haben ein Service Edge am Leaf 6

6.1 Vollständiges Script

Hier das ganze Script mit dem Namen wcord.py

```
# -*- coding: utf-8 -*-
from mininet.topo import Topo
from mininet.node import Node

class classRouter (Node):
    def config (self, **params):
              super (classRouter, self).config (**params)

              # Routing einschalten nachdem der Host hochgefahren ist
              self.cmd ('sysctl net.ipv4.ip_forward=1')

    def terminate (self):

              # Routing wieder abschalten bevor der Host runterfährt
              self.cmd ('sysctl net.ipv4.ip_forward=0')

              super (classRouter, self).terminate()

class classWCORD (Topo):
    def __init__ (self):
              Topo.__init__ (self)

              # Spine Switche
              spine1 = self.addSwitch ("spine1", dpid="1011");
              spine2 = self.addSwitch ("spine2", dpid="1012");
              spine3 = self.addSwitch ("spine3", dpid="1013");

              # Leaf Switche
              leaf1 = self.addSwitch ("leaf1", dpid="2021");
              leaf2 = self.addSwitch ("leaf2", dpid="2022");
              leaf3 = self.addSwitch ("leaf3", dpid="2023");
              leaf4 = self.addSwitch ("leaf4", dpid="2024");
              leaf5 = self.addSwitch ("leaf5", dpid="2025");
              leaf6 = self.addSwitch ("leaf6", dpid="2026");

              # Links zwischen Spine 1 und allen Leafs
              self.addLink (spine1, leaf1, 11, 1);
              self.addLink (spine1, leaf2, 12, 1);
              self.addLink (spine1, leaf3, 13, 1);
              self.addLink (spine1, leaf4, 14, 1);
              self.addLink (spine1, leaf5, 15, 1);
              self.addLink (spine1, leaf6, 16, 1);

              # Links zwischen Spine 2 und allen Leafs
              self.addLink (spine2, leaf1, 11, 2);
              self.addLink (spine2, leaf2, 12, 2);
              self.addLink (spine2, leaf3, 13, 2);
              self.addLink (spine2, leaf4, 14, 2);
              self.addLink (spine2, leaf5, 15, 2);
              self.addLink (spine2, leaf6, 16, 2);

              # Links zwischen Spine 3 und allen Leafs
              self.addLink (spine3, leaf1, 11, 3);
              self.addLink (spine3, leaf2, 12, 3);
              self.addLink (spine3, leaf3, 13, 3);
              self.addLink (spine3, leaf4, 14, 3);
              self.addLink (spine3, leaf5, 15, 3);
              self.addLink (spine3, leaf6, 16, 3);

              # Backbone Core Router cr<spine#><router#>
              cr11 = self.addHost ('cr11'); self.addLink (spine1, cr11, 1, 1);
              cr12 = self.addHost ('cr12'); self.addLink (spine1, cr12, 2, 1);

              cr21 = self.addHost ('cr21'); self.addLink (spine2, cr21, 1, 1);
              cr22 = self.addHost ('cr22'); self.addLink (spine2, cr22, 2, 1);

              cr31 = self.addHost ('cr31'); self.addLink (spine3, cr31, 1, 1);
              cr32 = self.addHost ('cr32'); self.addLink (spine3, cr32, 2, 1);

              # R-CORD DSLAM
              rdslam1 = self.addSwitch ("rdslam1", dpid="21011"); self.addLink (rdslam1, leaf1, 1, 11);
              rdslam2 = self.addSwitch ("rdslam2", dpid="21012"); self.addLink (rdslam2, leaf1, 1, 12);
              rdslam3 = self.addSwitch ("rdslam3", dpid="21013"); self.addLink (rdslam3, leaf1, 1, 13);

              # R-CORD DSL MODEM / CPEs
              dsl11 = self.addHost ('dsl11'); self.addLink (dsl11, rdslam1, 1, 11);
              dsl12 = self.addHost ('dsl12'); self.addLink (dsl12, rdslam1, 1, 12);
              dsl13 = self.addHost ('dsl13'); self.addLink (dsl13, rdslam1, 1, 13);

              dsl21 = self.addHost ('dsl21'); self.addLink (dsl21, rdslam2, 1, 11);
```

```
            dsl22 = self.addHost ('dsl22'); self.addLink (dsl22, rdslam2, 1, 12);
            dsl23 = self.addHost ('dsl23'); self.addLink (dsl23, rdslam2, 1, 13);

            dsl31 = self.addHost ('dsl31'); self.addLink (dsl31, rdslam3, 1, 11);
            dsl32 = self.addHost ('dsl32'); self.addLink (dsl32, rdslam3, 1, 12);
            dsl33 = self.addHost ('dsl33'); self.addLink (dsl33, rdslam3, 1, 13);

            # R-CORD OLT
            rolt1 = self.addSwitch ("rolt1", dpid="21021"); self.addLink (rolt1, leaf2, 1, 11);
            rolt2 = self.addSwitch ("rolt2", dpid="21022"); self.addLink (rolt2, leaf2, 1, 12);
            rolt3 = self.addSwitch ("rolt3", dpid="21023"); self.addLink (rolt3, leaf2, 1, 13);

            # R-CORD ONT MODEM / CPEs
            ont11 = self.addHost ('ont11'); self.addLink (ont11, rolt1, 1, 11);
            ont12 = self.addHost ('ont12'); self.addLink (ont12, rolt1, 1, 12);
            ont13 = self.addHost ('ont13'); self.addLink (ont13, rolt1, 1, 13);

            ont21 = self.addHost ('ont21'); self.addLink (ont21, rolt2, 1, 11);
            ont22 = self.addHost ('ont22'); self.addLink (ont22, rolt2, 1, 12);
            ont23 = self.addHost ('ont23'); self.addLink (ont23, rolt2, 1, 13);

            ont31 = self.addHost ('ont31'); self.addLink (ont31, rolt3, 1, 11);
            ont32 = self.addHost ('ont32'); self.addLink (ont32, rolt3, 1, 12);
            ont33 = self.addHost ('ont33'); self.addLink (ont33, rolt3, 1, 13);

            # R-CORD OLT/GFAST
            roltgf1 = self.addSwitch ("roltgf1", dpid="21031"); self.addLink (roltgf1, leaf3, 1, 11);
            roltgf2 = self.addSwitch ("roltgf2", dpid="21032"); self.addLink (roltgf2, leaf3, 1, 12);
            roltgf3 = self.addSwitch ("roltgf3", dpid="21033"); self.addLink (roltgf3, leaf3, 1, 13);

            # R-CORD GFAST
            rgf11 = self.addSwitch ("rgf11", dpid="31011"); self.addLink (rgf11, roltgf1, 1, 11);
            rgf12 = self.addSwitch ("rgf12", dpid="31012"); self.addLink (rgf12, roltgf1, 1, 12);
            rgf13 = self.addSwitch ("rgf13", dpid="31013"); self.addLink (rgf13, roltgf1, 1, 13);

            # R-CORD DSL MODEM / CPEs
            dslgf111 = self.addHost ('dslgf111'); self.addLink (dslgf111, rgf11, 1, 11);
            dslgf112 = self.addHost ('dslgf112'); self.addLink (dslgf112, rgf11, 1, 12);
            dslgf113 = self.addHost ('dslgf113'); self.addLink (dslgf113, rgf11, 1, 13);

            dslgf121 = self.addHost ('dslgf121'); self.addLink (dslgf121, rgf12, 1, 11);
            dslgf122 = self.addHost ('dslgf122'); self.addLink (dslgf122, rgf12, 1, 12);
            dslgf123 = self.addHost ('dslgf123'); self.addLink (dslgf123, rgf12, 1, 13);

            dslgf131 = self.addHost ('dslgf131'); self.addLink (dslgf131, rgf13, 1, 11);
            dslgf132 = self.addHost ('dslgf132'); self.addLink (dslgf132, rgf13, 1, 12);
            dslgf133 = self.addHost ('dslgf133'); self.addLink (dslgf133, rgf13, 1, 13);

            # R-CORD GFAST
            rgf21 = self.addSwitch ("rgf21", dpid="31021"); self.addLink (rgf21, roltgf2, 1, 11);
            rgf22 = self.addSwitch ("rgf22", dpid="31022"); self.addLink (rgf22, roltgf2, 1, 12);
            rgf23 = self.addSwitch ("rgf23", dpid="31023"); self.addLink (rgf23, roltgf2, 1, 13);

            # R-CORD DSL MODEM / CPEs
            dslgf211 = self.addHost ('dslgf211'); self.addLink (dslgf211, rgf21, 1, 11);
            dslgf212 = self.addHost ('dslgf212'); self.addLink (dslgf212, rgf21, 1, 12);
            dslgf213 = self.addHost ('dslgf213'); self.addLink (dslgf213, rgf21, 1, 13);

            dslgf221 = self.addHost ('dslgf221'); self.addLink (dslgf221, rgf22, 1, 11);
            dslgf222 = self.addHost ('dslgf222'); self.addLink (dslgf222, rgf22, 1, 12);
            dslgf223 = self.addHost ('dslgf223'); self.addLink (dslgf223, rgf22, 1, 13);

            dslgf231 = self.addHost ('dslgf231'); self.addLink (dslgf231, rgf23, 1, 11);
            dslgf232 = self.addHost ('dslgf232'); self.addLink (dslgf232, rgf23, 1, 12);
            dslgf233 = self.addHost ('dslgf233'); self.addLink (dslgf233, rgf23, 1, 13);

            # R-CORD GFAST
            rgf31 = self.addSwitch ("rgf31", dpid="31031"); self.addLink (rgf31, roltgf3, 1, 11);
            rgf32 = self.addSwitch ("rgf32", dpid="31032"); self.addLink (rgf32, roltgf3, 1, 12);
            rgf33 = self.addSwitch ("rgf33", dpid="31033"); self.addLink (rgf33, roltgf3, 1, 13);

            # R-CORD DSL MODEM / CPEs
            dslgf311 = self.addHost ('dslgf311'); self.addLink (dslgf311, rgf31, 1, 11);
            dslgf312 = self.addHost ('dslgf312'); self.addLink (dslgf312, rgf31, 1, 12);
            dslgf313 = self.addHost ('dslgf313'); self.addLink (dslgf313, rgf31, 1, 13);

            dslgf321 = self.addHost ('dslgf321'); self.addLink (dslgf321, rgf32, 1, 11);
            dslgf322 = self.addHost ('dslgf322'); self.addLink (dslgf322, rgf32, 1, 12);
            dslgf323 = self.addHost ('dslgf323'); self.addLink (dslgf323, rgf32, 1, 13);

            dslgf331 = self.addHost ('dslgf331'); self.addLink (dslgf331, rgf33, 1, 11);
            dslgf332 = self.addHost ('dslgf332'); self.addLink (dslgf332, rgf33, 1, 12);
            dslgf333 = self.addHost ('dslgf333'); self.addLink (dslgf333, rgf33, 1, 13);

            # M-CORD ENODEB
            menodeb1 = self.addSwitch ("menodeb1", dpid="41011"); self.addLink (menodeb1, leaf4, 1, 11);
            menodeb2 = self.addSwitch ("menodeb2", dpid="41012"); self.addLink (menodeb2, leaf4, 1, 12);
```

```
            menodeb3 = self.addSwitch ("menodeb3", dpid="41013"); self.addLink (menodeb3, leaf4, 1, 13);

            # M-CORD CELL PHONE
            cp11 = self.addHost ('cp11'); self.addLink (cp11, menodeb1, 1, 11);
            cp12 = self.addHost ('cp12'); self.addLink (cp12, menodeb1, 1, 12);
            cp13 = self.addHost ('cp13'); self.addLink (cp13, menodeb1, 1, 13);

            cp21 = self.addHost ('cp21'); self.addLink (cp21, menodeb2, 1, 11);
            cp22 = self.addHost ('cp22'); self.addLink (cp22, menodeb2, 1, 12);
            cp23 = self.addHost ('cp23'); self.addLink (cp23, menodeb2, 1, 13);

            cp31 = self.addHost ('cp31'); self.addLink (cp31, menodeb3, 1, 11);
            cp32 = self.addHost ('cp32'); self.addLink (cp32, menodeb3, 1, 12);
            cp33 = self.addHost ('cp33'); self.addLink (cp33, menodeb3, 1, 13);

            # E-CORD
            ecpe11 = self.addHost ('ecpe11'); self.addLink (ecpe11, leaf5, 1, 11);
            ecpe12 = self.addHost ('ecpe12'); self.addLink (ecpe12, leaf5, 1, 12);
            ecpe13 = self.addHost ('ecpe13'); self.addLink (ecpe13, leaf5, 1, 13);

            # Service Edge
            sbng11 = self.addHost ('sbng11'); self.addLink (sbng11, leaf6, 1, 11);
            sbng21 = self.addHost ('sbng21'); self.addLink (sbng21, leaf6, 1, 12);

            sdhcp11 = self.addHost ('sdhcp11'); self.addLink (sdhcp11, leaf6, 1, 13);
            sdhcp21 = self.addHost ('sdhcp21'); self.addLink (sdhcp21, leaf6, 1, 14);

            sdns11 = self.addHost ('sdns11'); self.addLink (sdns11, leaf6, 1, 15);
            sdns21 = self.addHost ('sdns21'); self.addLink (sdns21, leaf6, 1, 16);

            saaa11 = self.addHost ('saaa11'); self.addLink (saaa11, leaf6, 1, 17);
            saaa21 = self.addHost ('saaa21'); self.addLink (saaa21, leaf6, 1, 18);

            scache11 = self.addHost ('scache11'); self.addLink (scache11, leaf6, 1, 19);
            scache21 = self.addHost ('scache21'); self.addLink (scache21, leaf6, 1, 20);


topos = { 'WCORD': ( lambda: classWCORD() ) }
```

6.2 Mininet Start

```
whurst@sdn-mininet-01:~$ sudo mn --custom=wcord.py --topo=WCORD --controller=remote,ip=10.0.0.194
*** Creating network
*** Adding controller
*** Adding hosts:
cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23 dsl31
dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211 dslgf212
dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322 dslgf323
dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11 saaa21
sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
*** Adding switches:
leaf1 leaf2 leaf3 leaf4 leaf5 leaf6 menodeb1 menodeb2 menodeb3 rdslam1 rdslam2 rdslam3 rgf11 rgf12 rgf13 rgf21 rgf22
rgf23 rgf31 rgf32 rgf33 rolt1 rolt2 rolt3 roltgf1 roltgf2 roltgf3 spine1 spine2 spine3
*** Adding links:
(cp11, menodeb1) (cp12, menodeb1) (cp13, menodeb1) (cp21, menodeb2) (cp22, menodeb2) (cp23, menodeb2) (cp31, menodeb3)
(cp32, menodeb3) (cp33, menodeb3) (dsl11, rdslam1) (dsl12, rdslam1) (dsl13, rdslam1) (dsl21, rdslam2) (dsl22, rdslam2)
(dsl23, rdslam2) (dsl31, rdslam3) (dsl32, rdslam3) (dsl33, rdslam3) (dslgf111, rgf11) (dslgf112, rgf11) (dslgf113,
rgf11) (dslgf121, rgf12) (dslgf122, rgf12) (dslgf123, rgf12) (dslgf131, rgf13) (dslgf132, rgf13) (dslgf133, rgf13)
(dslgf211, rgf21) (dslgf212, rgf21) (dslgf213, rgf21) (dslgf221, rgf22) (dslgf222, rgf22) (dslgf223, rgf22) (dslgf231,
rgf23) (dslgf232, rgf23) (dslgf233, rgf23) (dslgf311, rgf31) (dslgf312, rgf31) (dslgf313, rgf31) (dslgf321, rgf32)
(dslgf322, rgf32) (dslgf323, rgf32) (dslgf331, rgf33) (dslgf332, rgf33) (dslgf333, rgf33) (ecpe11, leaf5) (ecpe12,
leaf5) (ecpe13, leaf5) (menodeb1, leaf4) (menodeb2, leaf4) (menodeb3, leaf4) (ont11, rolt1) (ont12, rolt1) (ont13,
rolt1) (ont21, rolt2) (ont22, rolt2) (ont23, rolt2) (ont31, rolt3) (ont32, rolt3) (ont33, rolt3) (rdslam1, leaf1)
(rdslam2, leaf1) (rdslam3, leaf1) (rgf11, roltgf1) (rgf12, roltgf1) (rgf13, roltgf1) (rgf21, roltgf2) (rgf22, roltgf2)
(rgf23, roltgf2) (rgf31, roltgf3) (rgf32, roltgf3) (rgf33, roltgf3) (rolt1, leaf2) (rolt2, leaf2) (rolt3, leaf2)
(roltgf1, leaf3) (roltgf2, leaf3) (roltgf3, leaf3) (saaa11, leaf6) (saaa21, leaf6) (sbng11, leaf6) (sbng21, leaf6)
(scache11, leaf6) (scache21, leaf6) (sdhcp11, leaf6) (sdhcp21, leaf6) (sdns11, leaf6) (sdns21, leaf6) (spine1, cr11)
(spine1, cr12) (spine1, leaf1) (spine1, leaf2) (spine1, leaf3) (spine1, leaf4) (spine1, leaf5) (spine1, leaf6)
(spine2, cr21) (spine2, cr22) (spine2, leaf1) (spine2, leaf2) (spine2, leaf3) (spine2, leaf4) (spine2, leaf5) (spine2,
leaf6) (spine3, cr31) (spine3, cr32) (spine3, leaf1) (spine3, leaf2) (spine3, leaf3) (spine3, leaf4) (spine3, leaf5)
(spine3, leaf6)
*** Configuring hosts
cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23 dsl31
dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211 dslgf212
dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322 dslgf323
dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11 saaa21
sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
*** Starting controller
c0
*** Starting 30 switches
leaf1 leaf2 leaf3 leaf4 leaf5 leaf6 menodeb1 menodeb2 menodeb3 rdslam1 rdslam2 rdslam3 rgf11 rgf12 rgf13 rgf21 rgf22
rgf23 rgf31 rgf32 rgf33 rolt1 rolt2 rolt3 roltgf1 roltgf2 roltgf3 spine1 spine2 spine3 ...
*** Starting CLI:
mininet> dump
<Host cp11: cp11-eth1:10.0.0.1 pid=26525>
<Host cp12: cp12-eth1:10.0.0.2 pid=26528>
<Host cp13: cp13-eth1:10.0.0.3 pid=26531>
<Host cp21: cp21-eth1:10.0.0.4 pid=26534>
<Host cp22: cp22-eth1:10.0.0.5 pid=26537>
<Host cp23: cp23-eth1:10.0.0.6 pid=26540>
<Host cp31: cp31-eth1:10.0.0.7 pid=26543>
<Host cp32: cp32-eth1:10.0.0.8 pid=26546>
<Host cp33: cp33-eth1:10.0.0.9 pid=26549>
<Host cr11: cr11-eth1:10.0.0.10 pid=26552>
<Host cr12: cr12-eth1:10.0.0.11 pid=26555>
<Host cr21: cr21-eth1:10.0.0.12 pid=26558>
<Host cr22: cr22-eth1:10.0.0.13 pid=26561>
<Host cr31: cr31-eth1:10.0.0.14 pid=26564>
<Host cr32: cr32-eth1:10.0.0.15 pid=26567>
<Host dsl11: dsl11-eth1:10.0.0.16 pid=26570>
<Host dsl12: dsl12-eth1:10.0.0.17 pid=26573>
<Host dsl13: dsl13-eth1:10.0.0.18 pid=26576>
<Host dsl21: dsl21-eth1:10.0.0.19 pid=26579>
<Host dsl22: dsl22-eth1:10.0.0.20 pid=26582>
<Host dsl23: dsl23-eth1:10.0.0.21 pid=26585>
<Host dsl31: dsl31-eth1:10.0.0.22 pid=26588>
<Host dsl32: dsl32-eth1:10.0.0.23 pid=26591>
<Host dsl33: dsl33-eth1:10.0.0.24 pid=26594>
<Host dslgf111: dslgf111-eth1:10.0.0.25 pid=26597>
<Host dslgf112: dslgf112-eth1:10.0.0.26 pid=26600>
<Host dslgf113: dslgf113-eth1:10.0.0.27 pid=26603>
<Host dslgf121: dslgf121-eth1:10.0.0.28 pid=26606>
<Host dslgf122: dslgf122-eth1:10.0.0.29 pid=26609>
<Host dslgf123: dslgf123-eth1:10.0.0.30 pid=26612>
<Host dslgf131: dslgf131-eth1:10.0.0.31 pid=26615>
<Host dslgf132: dslgf132-eth1:10.0.0.32 pid=26618>
<Host dslgf133: dslgf133-eth1:10.0.0.33 pid=26621>
<Host dslgf211: dslgf211-eth1:10.0.0.34 pid=26624>
<Host dslgf212: dslgf212-eth1:10.0.0.35 pid=26627>
<Host dslgf213: dslgf213-eth1:10.0.0.36 pid=26630>
<Host dslgf221: dslgf221-eth1:10.0.0.37 pid=26633>
<Host dslgf222: dslgf222-eth1:10.0.0.38 pid=26636>
<Host dslgf223: dslgf223-eth1:10.0.0.39 pid=26639>
<Host dslgf231: dslgf231-eth1:10.0.0.40 pid=26642>
<Host dslgf232: dslgf232-eth1:10.0.0.41 pid=26645>
<Host dslgf233: dslgf233-eth1:10.0.0.42 pid=26648>
<Host dslgf311: dslgf311-eth1:10.0.0.43 pid=26651>
<Host dslgf312: dslgf312-eth1:10.0.0.44 pid=26654>
<Host dslgf313: dslgf313-eth1:10.0.0.45 pid=26657>
<Host dslgf321: dslgf321-eth1:10.0.0.46 pid=26660>
<Host dslgf322: dslgf322-eth1:10.0.0.47 pid=26663>
<Host dslgf323: dslgf323-eth1:10.0.0.48 pid=26666>
<Host dslgf331: dslgf331-eth1:10.0.0.49 pid=26669>
<Host dslgf332: dslgf332-eth1:10.0.0.50 pid=26672>
<Host dslgf333: dslgf333-eth1:10.0.0.51 pid=26675>
<Host ecpe11: ecpe11-eth1:10.0.0.52 pid=26678>
<Host ecpe12: ecpe12-eth1:10.0.0.53 pid=26681>
<Host ecpe13: ecpe13-eth1:10.0.0.54 pid=26684>
<Host ont11: ont11-eth1:10.0.0.55 pid=26687>
```

```
<Host ont12: ont12-eth1:10.0.0.56 pid=26690>
<Host ont13: ont13-eth1:10.0.0.57 pid=26693>
<Host ont21: ont21-eth1:10.0.0.58 pid=26696>
<Host ont22: ont22-eth1:10.0.0.59 pid=26699>
<Host ont23: ont23-eth1:10.0.0.60 pid=26702>
<Host ont31: ont31-eth1:10.0.0.61 pid=26705>
<Host ont32: ont32-eth1:10.0.0.62 pid=26708>
<Host ont33: ont33-eth1:10.0.0.63 pid=26711>
<Host saaa11: saaa11-eth1:10.0.0.64 pid=26714>
<Host saaa21: saaa21-eth1:10.0.0.65 pid=26717>
<Host sbng11: sbng11-eth1:10.0.0.66 pid=26720>
<Host sbng21: sbng21-eth1:10.0.0.67 pid=26723>
<Host scache11: scache11-eth1:10.0.0.68 pid=26726>
<Host scache21: scache21-eth1:10.0.0.69 pid=26729>
<Host sdhcp11: sdhcp11-eth1:10.0.0.70 pid=26732>
<Host sdhcp21: sdhcp21-eth1:10.0.0.71 pid=26735>
<Host sdns11: sdns11-eth1:10.0.0.72 pid=26738>
<Host sdns21: sdns21-eth1:10.0.0.73 pid=26741>
<OVSSwitch leaf1: lo:127.0.0.1,leaf1-eth1:None,leaf1-eth2:None,leaf1-eth3:None,leaf1-eth11:None,leaf1-
eth12:None,leaf1-eth13:None pid=26747>
<OVSSwitch leaf2: lo:127.0.0.1,leaf2-eth1:None,leaf2-eth2:None,leaf2-eth3:None,leaf2-eth11:None,leaf2-
eth12:None,leaf2-eth13:None pid=26750>
<OVSSwitch leaf3: lo:127.0.0.1,leaf3-eth1:None,leaf3-eth2:None,leaf3-eth3:None,leaf3-eth11:None,leaf3-
eth12:None,leaf3-eth13:None pid=26753>
<OVSSwitch leaf4: lo:127.0.0.1,leaf4-eth1:None,leaf4-eth2:None,leaf4-eth3:None,leaf4-eth11:None,leaf4-
eth12:None,leaf4-eth13:None pid=26756>
<OVSSwitch leaf5: lo:127.0.0.1,leaf5-eth1:None,leaf5-eth2:None,leaf5-eth3:None,leaf5-eth11:None,leaf5-
eth12:None,leaf5-eth13:None pid=26759>
<OVSSwitch leaf6: lo:127.0.0.1,leaf6-eth1:None,leaf6-eth2:None,leaf6-eth3:None,leaf6-eth11:None,leaf6-
eth12:None,leaf6-eth13:None,leaf6-eth14:None,leaf6-eth15:None,leaf6-eth16:None,leaf6-eth17:None,leaf6-
eth18:None,leaf6-eth19:None,leaf6-eth20:None pid=26762>
<OVSSwitch menodeb1: lo:127.0.0.1,menodeb1-eth1:None,menodeb1-eth11:None,menodeb1-eth12:None,menodeb1-eth13:None
pid=26765>
<OVSSwitch menodeb2: lo:127.0.0.1,menodeb2-eth1:None,menodeb2-eth11:None,menodeb2-eth12:None,menodeb2-eth13:None
pid=26768>
<OVSSwitch menodeb3: lo:127.0.0.1,menodeb3-eth1:None,menodeb3-eth11:None,menodeb3-eth12:None,menodeb3-eth13:None
pid=26771>
<OVSSwitch rdslam1: lo:127.0.0.1,rdslam1-eth1:None,rdslam1-eth11:None,rdslam1-eth12:None,rdslam1-eth13:None pid=26774>
<OVSSwitch rdslam2: lo:127.0.0.1,rdslam2-eth1:None,rdslam2-eth11:None,rdslam2-eth12:None,rdslam2-eth13:None pid=26777>
<OVSSwitch rdslam3: lo:127.0.0.1,rdslam3-eth1:None,rdslam3-eth11:None,rdslam3-eth12:None,rdslam3-eth13:None pid=26780>
<OVSSwitch rgf11: lo:127.0.0.1,rgf11-eth1:None,rgf11-eth11:None,rgf11-eth12:None,rgf11-eth13:None pid=26783>
<OVSSwitch rgf12: lo:127.0.0.1,rgf12-eth1:None,rgf12-eth11:None,rgf12-eth12:None,rgf12-eth13:None pid=26786>
<OVSSwitch rgf13: lo:127.0.0.1,rgf13-eth1:None,rgf13-eth11:None,rgf13-eth12:None,rgf13-eth13:None pid=26789>
<OVSSwitch rgf21: lo:127.0.0.1,rgf21-eth1:None,rgf21-eth11:None,rgf21-eth12:None,rgf21-eth13:None pid=26792>
<OVSSwitch rgf22: lo:127.0.0.1,rgf22-eth1:None,rgf22-eth11:None,rgf22-eth12:None,rgf22-eth13:None pid=26795>
<OVSSwitch rgf23: lo:127.0.0.1,rgf23-eth1:None,rgf23-eth11:None,rgf23-eth12:None,rgf23-eth13:None pid=26798>
<OVSSwitch rgf31: lo:127.0.0.1,rgf31-eth1:None,rgf31-eth11:None,rgf31-eth12:None,rgf31-eth13:None pid=26801>
<OVSSwitch rgf32: lo:127.0.0.1,rgf32-eth1:None,rgf32-eth11:None,rgf32-eth12:None,rgf32-eth13:None pid=26804>
<OVSSwitch rgf33: lo:127.0.0.1,rgf33-eth1:None,rgf33-eth11:None,rgf33-eth12:None,rgf33-eth13:None pid=26807>
<OVSSwitch rolt1: lo:127.0.0.1,rolt1-eth1:None,rolt1-eth11:None,rolt1-eth12:None,rolt1-eth13:None pid=26810>
<OVSSwitch rolt2: lo:127.0.0.1,rolt2-eth1:None,rolt2-eth11:None,rolt2-eth12:None,rolt2-eth13:None pid=26813>
<OVSSwitch rolt3: lo:127.0.0.1,rolt3-eth1:None,rolt3-eth11:None,rolt3-eth12:None,rolt3-eth13:None pid=26816>
<OVSSwitch roltgf1: lo:127.0.0.1,roltgf1-eth1:None,roltgf1-eth11:None,roltgf1-eth12:None,roltgf1-eth13:None pid=26819>
<OVSSwitch roltgf2: lo:127.0.0.1,roltgf2-eth1:None,roltgf2-eth11:None,roltgf2-eth12:None,roltgf2-eth13:None pid=26822>
<OVSSwitch roltgf3: lo:127.0.0.1,roltgf3-eth1:None,roltgf3-eth11:None,roltgf3-eth12:None,roltgf3-eth13:None pid=26825>
<OVSSwitch spine1: lo:127.0.0.1,spine1-eth1:None,spine1-eth2:None,spine1-eth11:None,spine1-eth12:None,spine1-
eth13:None,spine1-eth14:None,spine1-eth15:None,spine1-eth16:None pid=26828>
<OVSSwitch spine2: lo:127.0.0.1,spine2-eth1:None,spine2-eth2:None,spine2-eth11:None,spine2-eth12:None,spine2-
eth13:None,spine2-eth14:None,spine2-eth15:None,spine2-eth16:None pid=26831>
<OVSSwitch spine3: lo:127.0.0.1,spine3-eth1:None,spine3-eth2:None,spine3-eth11:None,spine3-eth12:None,spine3-
eth13:None,spine3-eth14:None,spine3-eth15:None,spine3-eth16:None pid=26834>
<RemoteController{'ip': '10.0.0.194'} c0: 10.0.0.194:6633 pid=26519>
```

6.3 Ping all

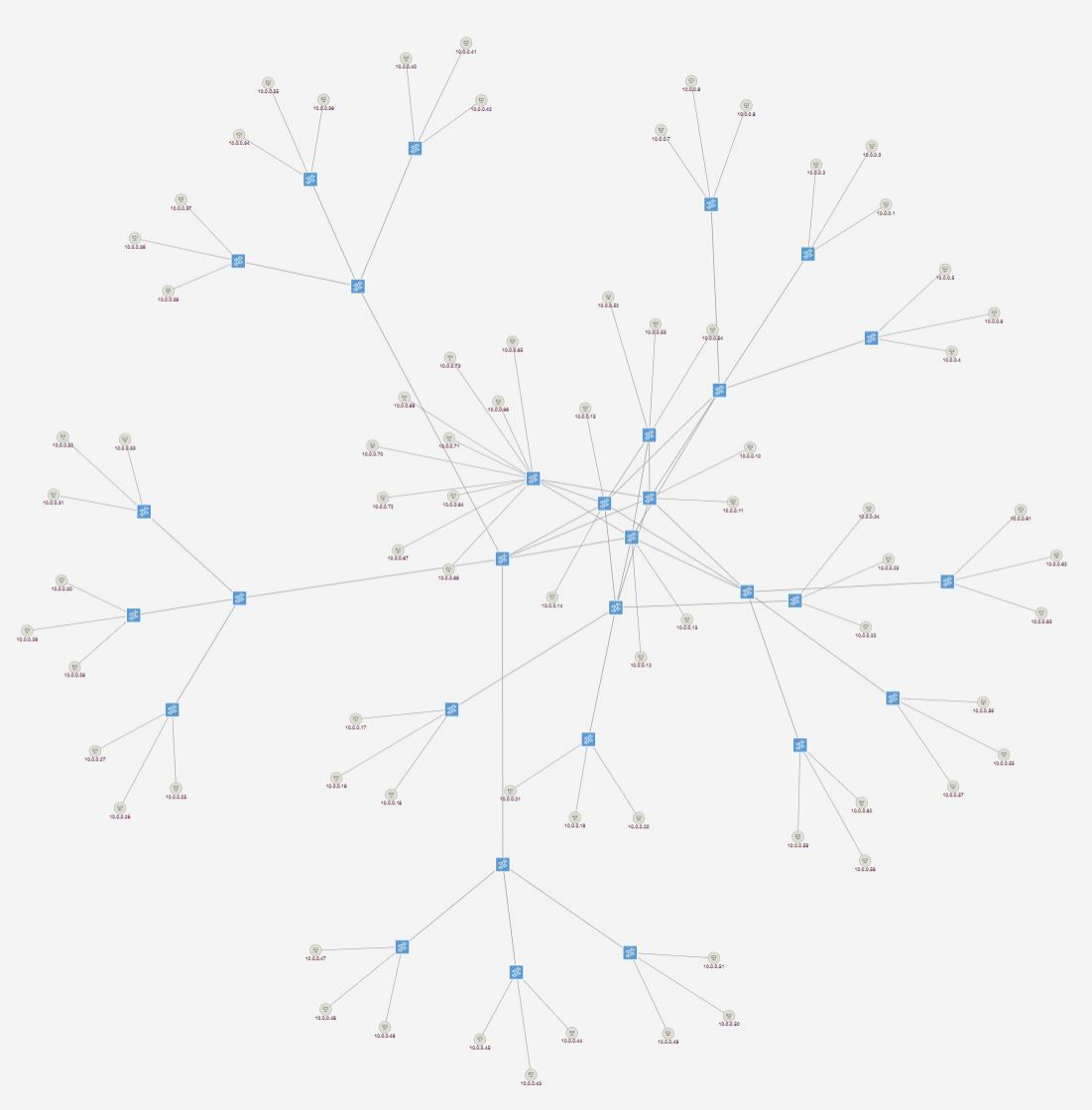
Nun … ein Ping all dauert nun etwas …

```
mininet> pingall
*** Ping: testing ping reachability
cp11 -> cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23
dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211 dslgf212
dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322 dslgf323
dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11 saaa21
sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
cp12 -> cp11 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23
dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211 dslgf212
dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322 dslgf323
dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11 saaa21
sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
cp13 -> cp11 cp12 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23
dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211 dslgf212
dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322 dslgf323
dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11 saaa21
sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
cp21 -> cp11 cp12 cp13 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23
dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211 dslgf212
dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322 dslgf323
dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11 saaa21
sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
cp22 -> cp11 cp12 cp13 cp21 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23
dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211 dslgf212
dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322 dslgf323
dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11 saaa21
sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
cp23 -> cp11 cp12 cp13 cp21 cp22 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23


Ein „paar“ Zeilen entfernt


sbng11 -> cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22
dsl23 dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211
dslgf212 dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322
dslgf323 dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11
saaa21 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
sbng21 -> cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22
dsl23 dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211
dslgf212 dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322
dslgf323 dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11
saaa21 sbng11 scache11 scache21 sdhcp11 sdhcp21 sdns11 sdns21
scache11 -> cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22
dsl23 dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211
dslgf212 dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322
dslgf323 dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11
saaa21 sbng11 sbng21 scache21 sdhcp11 sdhcp21 sdns11 sdns21
scache21 -> cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22
dsl23 dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211
dslgf212 dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322
dslgf323 dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11
saaa21 sbng11 sbng21 scache11 sdhcp11 sdhcp21 sdns11 sdns21
sdhcp11 -> cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22
dsl23 dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211
dslgf212 dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322
dslgf323 dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11
saaa21 sbng11 sbng21 scache11 scache21 sdhcp21 sdns11 sdns21
sdhcp21 -> cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22
dsl23 dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211
dslgf212 dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322
dslgf323 dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11
saaa21 sbng11 sbng21 scache11 scache21 sdhcp11 sdns11 sdns21
sdns11 -> cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22
dsl23 dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211
dslgf212 dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322
dslgf323 dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11
saaa21 sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11
sdns21 -> cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22
dsl23 dsl31 dsl32 dsl33 dslgf111 dslgf112 dslgf113 dslgf121 dslgf122 dslgf123 dslgf131 dslgf132 dslgf133 dslgf211
dslgf212 dslgf213 dslgf221 dslgf222 dslgf223 dslgf231 dslgf232 dslgf233 dslgf311 dslgf312 dslgf313 dslgf321 dslgf322
dslgf323 dslgf331 dslgf332 dslgf333 ecpe11 ecpe12 ecpe13 ont11 ont12 ont13 ont21 ont22 ont23 ont31 ont32 ont33 saaa11
saaa21 sbng11 sbng21 scache11 scache21 sdhcp11 sdhcp21 sdns11
*** Results: 0% dropped (5256/5256 received)
mininet>
```

Immerhin … 5256 und 0% verloren. Fein.

6.4 ONOS Topologie View

Das ganze sieht im ONOS dann sehr seltsam strukturiert dann so aus …



Mit etwas Fanta-Sie geht das … oder auch nicht …

7 Konfiguration in ONOS

7.1 lè Problem

Das Problem was wir jetzt haben, ist das wir mit Mininet dem ONOS ein Netzwerk unter die Nase reiben, was ja so im normalen Leben nicht vorkommt. In der Realität baut jemand ein Switch in ein Rack und erwartet das ONOS den kennt.

Sprich, wir müssten eigentlich ONOS konfigurieren und nicht Mininet. Das stellt uns jetzt vor das Problem das wir jetzt zwei Konfigurationen pflegen müssten und diese auch noch Synchron halten müssten. Das macht natürlich kein Sinn.

Aus diesem Grunde werden wir das Mininet Script aufbohren und eine ONOS Mund gerechte Netz Konfiguration erzeugen, die wir dann einfach in das ONOS importieren können. Damit verbleiben alle Informationen an einer Stelle.

7.2 dé Lösung

Nach jedem `addSwitch` oder `addHost` rufen wir eine Funktion auf, die uns für ONOS ein JSON generiert mit den passenden Werten. Das ganze speichern wir eine Zeichenkette. Am Ende des Scripts werden wir das ganze JSON inklusive des REST API Aufrufs mit `curl` ausgeben. In einem anderen Terminal kann man das dann einfach in der CLI ausführen, oder speichern.

Zugegeben, etwas komisch, aber super praktisch.

7.3 Änderungen am Script

Wir brauchen kleine Helfer Funktionen die uns das JSON erstellen.

```
        def convertCellX (self, x):
                return ( x * 2 );

        def convertCellY (self, y):
                return (( y * 4 ) * - 1);

        def outputOnosNetcfgSwitch (self, name, id, x, y):
                return '"of:00000000000' + id + '" : { "basic": { "name":"' + name + '",
"latitude":' + str(self.convertCellY(y)) + ', "longitude":' + str(self.convertCellX(x)) + ' }
},'

        def outputOnosNetcfgHost (self, name, mac, x, y):
                theMAC = mac;
                while (len (theMAC) < 17):
                        tempx = "00:" + theMAC;
                        theMAC = tempx;

                return '"' + theMAC + '/-1" : { "basic": { "name":"' + name + '", "latitude":'
+ str(self.convertCellY(y)) + ', "longitude":' + str(self.convertCellX(x)) + ' } },'
```

Die Funktionen rufen wir dann nach der Erstellung auf und das JSON speichern wir in einem großen String

```
                spine1 = self.addSwitch ("spine1", dpid="1011");
                spine2 = self.addSwitch ("spine2", dpid="1012");
                spine3 = self.addSwitch ("spine3", dpid="1013");

                onosSW += self.outputOnosNetcfgSwitch ("SPINE 1", "01011", 13, 5);
                onosSW += self.outputOnosNetcfgSwitch ("SPINE 2", "01012", 43, 5);
                onosSW += self.outputOnosNetcfgSwitch ("SPINE 3", "01013", 73, 5);
```

Und für Hosts brauchen wir dann noch eine MAC Adresse, sonst findet die ONOS nicht

```
                dsl11 = self.addHost ('dsl11', mac='00:00:00:01:01:01');
                dsl12 = self.addHost ('dsl12', mac='00:00:00:02:01:01');
                dsl13 = self.addHost ('dsl13', mac='00:00:00:03:01:01');

                onosH += self.outputOnosNetcfgHost ("dsl11", "01:01:01", 2, 11);
                onosH += self.outputOnosNetcfgHost ("dsl12", "02:01:01", 3, 11);
                onosH += self.outputOnosNetcfgHost ("dsl13", "03:01:01", 4, 11);
```

Am Ende geben wir das ganze Copy&Paste gerecht einfach aus

```
                print 'curl --user onos:rocks -X POST -H "Content-Type: application/json"
http://sdn-onos-dev-01:8181/onos/v1/network/configuration/ -d \'';
                print '{';
                print '"devices" : {';
                print onosSW;
                print '"of:0000000000000000" : { "basic": { "name":"NOT_EXISTENT" } }';
                print '},';
                print '"hosts" : {';
                print onosH;
                print '"00:00:00:00:00:00/-1" : { "basic": { "name":"NOT_EXISTENT" } }';
                print '}';
                print '}';
                print '\'';
```

Wenn man nun Mininet startet, wird das gesamte JSON ausgegeben

```
whurst@sdn-mininet-01:~$ sudo mn --custom=wcord.py --topo=WCORD --controller=remote,ip=10.0.0.194
curl --user onos:rocks -X POST -H "Content-Type: application/json" http://sdn-onos-dev-
01:8181/onos/v1/network/configuration/ -d '
{
"devices" : {
"of:0000000000001011" : { "basic": { "name":"SPINE 1", "latitude":-20, "longitude":26 } },"of:0000000000001012" : {
"basic": { "name":"SPINE 2", "latitude":-20, "longitude":86 } },"of:0000000000001013" : { "basic": { "name":"SPINE 3",
"latitude":-20, "longitude":146 } },"of:0000000000002021" : { "basic": { "name":"LEAF 1", "latitude":-28,
"longitude":14 } },"of:0000000000002022" : { "basic": { "name":"LEAF 2", "latitude":-28, "longitude":38 }
},"of:0000000000002023" : { "basic": { "name":"LEAF 3", "latitude":-28, "longitude":86 } },"of:0000000000002024" : {
"basic": { "name":"LEAF 4", "latitude":-28, "longitude":134 } },"of:0000000000002025" : { "basic": { "name":"LEAF 5",
"latitude":-28, "longitude":152 } },"of:0000000000002026" : { "basic": { "name":"LEAF 6", "latitude":-28,
"longitude":170 } },"of:0000000000021011" : { "basic": { "name":"DSLAM 1", "latitude":-36, "longitude":6 }
},"of:0000000000021012" : { "basic": { "name":"DSLAM 2", "latitude":-36, "longitude":14 } },"of:0000000000021013" : {
"basic": { "name":"DSLAM 3", "latitude":-36, "longitude":22 } },"of:0000000000021021" : { "basic": { "name":"OLT 1",
"latitude":-36, "longitude":30 } },"of:0000000000021022" : { "basic": { "name":"OLT 2", "latitude":-36, "longitude":38
} },"of:0000000000021023" : { "basic": { "name":"OLT 3", "latitude":-36, "longitude":46 } },"of:0000000000021031" : {
"basic": { "name":"OLT GF 1", "latitude":-36, "longitude":62 } },"of:0000000000021032" : { "basic": { "name":"OLT GF
2", "latitude":-36, "longitude":86 } },"of:0000000000021033" : { "basic": { "name":"OLT GF 3", "latitude":-36,
"longitude":110 } },"of:0000000000031011" : { "basic": { "name":"GF 11", "latitude":-44, "longitude":54 }
},"of:0000000000031012" : { "basic": { "name":"GF 12", "latitude":-44, "longitude":62 } },"of:0000000000031013" : {
"basic": { "name":"GF 13", "latitude":-44, "longitude":70 } },"of:0000000000031021" : { "basic": { "name":"GF 21",
"latitude":-44, "longitude":78 } },"of:0000000000031022" : { "basic": { "name":"GF 22", "latitude":-44, "longitude":86
} },"of:0000000000031023" : { "basic": { "name":"GF 23", "latitude":-44, "longitude":94 } },"of:0000000000031031" : {
"basic": { "name":"GF 33", "latitude":-44, "longitude":102 } },"of:0000000000031032" : { "basic": { "name":"GF 33",
"latitude":-44, "longitude":110 } },"of:0000000000031033" : { "basic": { "name":"GF 33", "latitude":-44,
"longitude":118 } },"of:0000000000041011" : { "basic": { "name":"eNodeB 1", "latitude":-36, "longitude":126 }
},"of:0000000000041012" : { "basic": { "name":"eNodeB 2", "latitude":-36, "longitude":134 } },"of:0000000000041013" :
{ "basic": { "name":"eNodeB 3", "latitude":-36, "longitude":142 } },
"of:0000000000000000" : { "basic": { "name":"NOT_EXISTENT" } }
},
"hosts" : {
"00:00:00:01:01:69/-1" : { "basic": { "name":"CR 11", "latitude":-12, "longitude":24 } },"00:00:00:02:01:69/-1" : {
"basic": { "name":"CR 12", "latitude":-12, "longitude":28 } },"00:00:00:01:02:69/-1" : { "basic": { "name":"CR 21",
"latitude":-12, "longitude":84 } },"00:00:00:02:02:69/-1" : { "basic": { "name":"CR 22", "latitude":-12,
"longitude":88 } },"00:00:00:01:03:69/-1" : { "basic": { "name":"CR 31", "latitude":-12, "longitude":144 }
},"00:00:00:02:03:69/-1" : { "basic": { "name":"CR 32", "latitude":-12, "longitude":148 } },"00:00:00:01:01:01/-1" : {
"basic": { "name":"dsl11", "latitude":-44, "longitude":4 } },"00:00:00:02:01:01/-1" : { "basic": { "name":"dsl12",
"latitude":-44, "longitude":6 } },"00:00:00:03:01:01/-1" : { "basic": { "name":"dsl13", "latitude":-44, "longitude":8
} },"00:00:00:01:02:01/-1" : { "basic": { "name":"dsl21", "latitude":-44, "longitude":12 } },"00:00:00:02:02:01/-1" :
{ "basic": { "name":"dsl22", "latitude":-44, "longitude":14 } },"00:00:00:03:02:01/-1" : { "basic": { "name":"dsl23",
"latitude":-44, "longitude":16 } },"00:00:00:01:03:01/-1" : { "basic": { "name":"dsl31", "latitude":-44,
"longitude":20 } },"00:00:00:02:03:01/-1" : { "basic": { "name":"dsl32", "latitude":-44, "longitude":22 }
},"00:00:00:03:03:01/-1" : { "basic": { "name":"dsl33", "latitude":-44, "longitude":24 } },"00:00:00:01:01:02/-1" : {
"basic": { "name":"ont11", "latitude":-44, "longitude":28 } },"00:00:00:02:01:02/-1" : { "basic": { "name":"ont12",
"latitude":-44, "longitude":30 } },"00:00:00:03:01:02/-1" : { "basic": { "name":"ont13", "latitude":-44,
"longitude":32 } },"00:00:00:01:02:02/-1" : { "basic": { "name":"ont21", "latitude":-44, "longitude":36 }
},"00:00:00:02:02:02/-1" : { "basic": { "name":"ont22", "latitude":-44, "longitude":38 } },"00:00:00:03:02:02/-1" : {
"basic": { "name":"ont23", "latitude":-44, "longitude":40 } },"00:00:00:01:03:02/-1" : { "basic": { "name":"ont31",
"latitude":-44, "longitude":44 } },"00:00:00:02:03:02/-1" : { "basic": { "name":"ont32", "latitude":-44,
"longitude":46 } },"00:00:00:03:03:02/-1" : { "basic": { "name":"ont33", "latitude":-44, "longitude":48 }
```

```
},"00:00:01:01:01:03/-1" : { "basic": { "name":"dslgf111", "latitude":-52, "longitude":52 } },"00:00:02:01:01:03/-1" :
{ "basic": { "name":"dslgf112", "latitude":-52, "longitude":54 } },"00:00:03:01:01:03/-1" : { "basic": {
"name":"dslgf113", "latitude":-52, "longitude":56 } },"00:00:01:02:01:03/-1" : { "basic": { "name":"dslgf121",
"latitude":-52, "longitude":60 } },"00:00:02:02:01:03/-1" : { "basic": { "name":"dslgf122", "latitude":-52,
"longitude":62 } },"00:00:03:02:01:03/-1" : { "basic": { "name":"dslgf123", "latitude":-52, "longitude":64 }
},"00:00:01:03:01:03/-1" : { "basic": { "name":"dslgf131", "latitude":-52, "longitude":68 } },"00:00:02:03:01:03/-1" :
{ "basic": { "name":"dslgf132", "latitude":-52, "longitude":70 } },"00:00:03:03:01:03/-1" : { "basic": {
"name":"dslgf133", "latitude":-52, "longitude":72 } },"00:00:01:01:02:03/-1" : { "basic": { "name":"dslgf211",
"latitude":-52, "longitude":76 } },"00:00:02:01:02:03/-1" : { "basic": { "name":"dslgf212", "latitude":-52,
"longitude":78 } },"00:00:03:01:02:03/-1" : { "basic": { "name":"dslgf213", "latitude":-52, "longitude":80 }
},"00:00:01:02:02:03/-1" : { "basic": { "name":"dslgf221", "latitude":-52, "longitude":84 } },"00:00:02:02:02:03/-1" :
{ "basic": { "name":"dslgf222", "latitude":-52, "longitude":86 } },"00:00:03:02:02:03/-1" : { "basic": {
"name":"dslgf223", "latitude":-52, "longitude":88 } },"00:00:01:03:02:03/-1" : { "basic": { "name":"dslgf231",
"latitude":-52, "longitude":92 } },"00:00:02:03:02:03/-1" : { "basic": { "name":"dslgf232", "latitude":-52,
"longitude":94 } },"00:00:03:03:02:03/-1" : { "basic": { "name":"dslgf233", "latitude":-52, "longitude":96 }
},"00:00:01:01:03:03/-1" : { "basic": { "name":"dslgf311", "latitude":-52, "longitude":100 } },"00:00:02:01:03:03/-1"
: { "basic": { "name":"dslgf312", "latitude":-52, "longitude":102 } },"00:00:03:01:03:03/-1" : { "basic": {
"name":"dslgf313", "latitude":-52, "longitude":104 } },"00:00:01:02:03:03/-1" : { "basic": { "name":"dslgf321",
"latitude":-52, "longitude":108 } },"00:00:02:02:03:03/-1" : { "basic": { "name":"dslgf322", "latitude":-52,
"longitude":110 } },"00:00:03:02:03:03/-1" : { "basic": { "name":"dslgf323", "latitude":-52, "longitude":112 }
},"00:00:01:03:03:03/-1" : { "basic": { "name":"dslgf331", "latitude":-52, "longitude":116 } },"00:00:02:03:03:03/-1"
: { "basic": { "name":"dslgf332", "latitude":-52, "longitude":118 } },"00:00:03:03:03:03/-1" : { "basic": {
"name":"dslgf333", "latitude":-52, "longitude":120 } },"00:00:00:01:01:04/-1" : { "basic": { "name":"cp11",
"latitude":-44, "longitude":124 } },"00:00:00:02:01:04/-1" : { "basic": { "name":"cp12", "latitude":-44,
"longitude":126 } },"00:00:00:03:01:04/-1" : { "basic": { "name":"cp13", "latitude":-44, "longitude":128 }
},"00:00:00:01:02:04/-1" : { "basic": { "name":"cp21", "latitude":-44, "longitude":132 } },"00:00:00:02:02:04/-1" : {
"basic": { "name":"cp22", "latitude":-44, "longitude":134 } },"00:00:00:03:02:04/-1" : { "basic": { "name":"cp23",
"latitude":-44, "longitude":136 } },"00:00:00:01:03:04/-1" : { "basic": { "name":"cp31", "latitude":-44,
"longitude":140 } },"00:00:00:02:03:04/-1" : { "basic": { "name":"cp32", "latitude":-44, "longitude":142 }
},"00:00:00:03:03:04/-1" : { "basic": { "name":"cp33", "latitude":-44, "longitude":144 } },"00:00:00:01:01:05/-1" : {
"basic": { "name":"ecpe11", "latitude":-36, "longitude":148 } },"00:00:00:02:01:05/-1" : { "basic": { "name":"ecpe12",
"latitude":-36, "longitude":152 } },"00:00:00:03:01:05/-1" : { "basic": { "name":"ecpe13", "latitude":-36,
"longitude":156 } },"00:00:00:01:01:06/-1" : { "basic": { "name":"sbng11", "latitude":-36, "longitude":160 }
},"00:00:00:02:01:06/-1" : { "basic": { "name":"sbng21", "latitude":-36, "longitude":162 } },"00:00:00:01:02:06/-1" :
{ "basic": { "name":"sdhcp11", "latitude":-44, "longitude":164 } },"00:00:00:02:02:06/-1" : { "basic": {
"name":"sdhcp21", "latitude":-44, "longitude":166 } },"00:00:00:01:03:06/-1" : { "basic": { "name":"sdns11",
"latitude":-52, "longitude":168 } },"00:00:00:02:03:06/-1" : { "basic": { "name":"sdns21", "latitude":-52,
"longitude":172 } },"00:00:00:01:04:06/-1" : { "basic": { "name":"saaa11", "latitude":-44, "longitude":174 }
},"00:00:00:02:04:06/-1" : { "basic": { "name":"saaa21", "latitude":-44, "longitude":176 } },"00:00:00:01:05:06/-1" :
{ "basic": { "name":"scache11", "latitude":-36, "longitude":178 } },"00:00:00:02:05:06/-1" : { "basic": {
"name":"scache21", "latitude":-36, "longitude":180 } },
"00:00:00:00:00:00/-1" : { "basic": { "name":"NOT_EXISTENT" } }
}
}
'
*** Creating network
*** Adding controller
*** Adding hosts:
cp11 cp12 cp13 cp21 cp22 cp23 cp31 cp32 cp33 cr11 cr12 cr21 cr22 cr31 cr32 dsl11 dsl12 dsl13 dsl21 dsl22 dsl23 dsl31
dsl32 dsl33 dslgf111
```

Jetzt kann man den Block (in Fett) in die Zwischenablage werfen und in einem anderen Terminal es 1:1 so ausführen. Das JSON wird dann dem ONOS gegeben und er baut uns dann alles fein.

7.4 Koordinaten

Um ein System im Raum zu positionieren benutzt ONOS geographische Längen und Breitengrade. Das ist zwar toll, weil man auch Landkarten in den Graph einzeichnen lassen kann. Hilft uns aber jetzt nicht direkt. Wir rechnen einfach Punkt Koordinaten um. Ich habe dazu eine Excel Tabelle erstellt mit allen Nodes und lese dann einfach die X und Y Koordinaten ab. Die beiden Funktionen `convertCellX` und `convertCellY` skalieren die Werte dann so wie wir man sie braucht. Man kann also die Höhe und Breite relativ schnell ändern in den beiden Funktionen.

Hier ein Beispiel meiner Excel Hilfe Netzplan Koordinaten Tabelle

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
2																																	
3												cr11		cr12																			
4																																	
5													spine 1																				
6																																	
7							leaf 1												leaf 2														
8																																	
9			dslam1				dslam2				dslam3				olt1				olt2				olt3								oltgf1		
10																																	
11		dsl11	dsl12	dsl13		dsl21	dsl22	dsl23		dsl31	dsl32	dsl33		ont11	ont12	ont13		ont21	ont22	ont23		ont31	ont32	ont33			gf11				gf12		
12																																	
13																										dslgf11	dslfg1	dslfg113		dslgf1	dslgf1	dlsgf123	
14																																	

7.5 Vollständiges Script

Hier noch einmal zur Sicherheit das gesamte Script, so wie es 100% funktioniert

```
# -*- coding: utf-8 -*-
from mininet.topo import Topo
from mininet.node import Node

class classRouter (Node):
    def config (self, **params):
        super (classRouter, self).config (**params)

        # Routing einschalten nachdem der Host hochgefahren ist
        self.cmd ('sysctl net.ipv4.ip_forward=1')

    def terminate (self):

        # Routing wieder abschalten bevor der Host runterfährt
        self.cmd ('sysctl net.ipv4.ip_forward=0')

        super (classRouter, self).terminate()

class classWCORD (Topo):

        def convertCellX (self, x):
                return ( x * 2 );

        def convertCellY (self, y):
                return (( y * 4 ) * - 1);

        def outputOnosNetcfgSwitch (self, name, id, x, y):
                return '"of:00000000000' + id + '" : { "basic": { "name":"' + name + '",
"latitude":' + str(self.convertCellY(y)) + ', "longitude":' + str(self.convertCellX(x)) + ' }
},'

        def outputOnosNetcfgHost (self, name, mac, x, y):
                theMAC = mac;
                while (len (theMAC) < 17):
                        tempx = "00:" + theMAC;
                        theMAC = tempx;

                return '"' + theMAC + '/-1" : { "basic": { "name":"' + name + '", "latitude":'
+ str(self.convertCellY(y)) + ', "longitude":' + str(self.convertCellX(x)) + ' } },'

        def __init__ (self):
                Topo.__init__ (self)

                onosSW = "";
                onosH = "";

                # Spine Switche
                spine1 = self.addSwitch ("spine1", dpid="1011");
                spine2 = self.addSwitch ("spine2", dpid="1012");
                spine3 = self.addSwitch ("spine3", dpid="1013");

                onosSW += self.outputOnosNetcfgSwitch ("SPINE 1", "01011", 13, 5);
                onosSW += self.outputOnosNetcfgSwitch ("SPINE 2", "01012", 43, 5);
                onosSW += self.outputOnosNetcfgSwitch ("SPINE 3", "01013", 73, 5);

                # Leaf Switche
                leaf1 = self.addSwitch ("leaf1", dpid="2021");
                leaf2 = self.addSwitch ("leaf2", dpid="2022");
                leaf3 = self.addSwitch ("leaf3", dpid="2023");
                leaf4 = self.addSwitch ("leaf4", dpid="2024");
                leaf5 = self.addSwitch ("leaf5", dpid="2025");
                leaf6 = self.addSwitch ("leaf6", dpid="2026");

                onosSW += self.outputOnosNetcfgSwitch ("LEAF 1", "02021", 7, 7);
                onosSW += self.outputOnosNetcfgSwitch ("LEAF 2", "02022", 19, 7);
                onosSW += self.outputOnosNetcfgSwitch ("LEAF 3", "02023", 43, 7);
                onosSW += self.outputOnosNetcfgSwitch ("LEAF 4", "02024", 67, 7);
                onosSW += self.outputOnosNetcfgSwitch ("LEAF 5", "02025", 76, 7);
                onosSW += self.outputOnosNetcfgSwitch ("LEAF 6", "02026", 85, 7);

                # Links zwischen Spine 1 und allen Leafs
                self.addLink (spine1, leaf1, 11, 1);
                self.addLink (spine1, leaf2, 12, 1);
                self.addLink (spine1, leaf3, 13, 1);
                self.addLink (spine1, leaf4, 14, 1);
```

```
                self.addLink (spine1, leaf5, 15, 1);
                self.addLink (spine1, leaf6, 16, 1);

                # Links zwischen Spine 2 und allen Leafs
                self.addLink (spine2, leaf1, 11, 2);
                self.addLink (spine2, leaf2, 12, 2);
                self.addLink (spine2, leaf3, 13, 2);
                self.addLink (spine2, leaf4, 14, 2);
                self.addLink (spine2, leaf5, 15, 2);
                self.addLink (spine2, leaf6, 16, 2);

                # Links zwischen Spine 3 und allen Leafs
                self.addLink (spine3, leaf1, 11, 3);
                self.addLink (spine3, leaf2, 12, 3);
                self.addLink (spine3, leaf3, 13, 3);
                self.addLink (spine3, leaf4, 14, 3);
                self.addLink (spine3, leaf5, 15, 3);
                self.addLink (spine3, leaf6, 16, 3);

                # Backbone Core Router cr<spine#><router#>
                # MAC 00:00:00:<host#>:<spine#>:69
                cr11 = self.addHost ('cr11', mac='00:00:00:01:01:69');
                cr12 = self.addHost ('cr12', mac='00:00:00:02:01:69');

                onosH += self.outputOnosNetcfgHost ("CR 11", "01:01:69", 12, 3);
                onosH += self.outputOnosNetcfgHost ("CR 12", "02:01:69", 14, 3);

                self.addLink (spine1, cr11, 1, 1);
                self.addLink (spine1, cr12, 2, 1);

                cr21 = self.addHost ('cr21', mac='00:00:00:01:02:69');
                cr22 = self.addHost ('cr22', mac='00:00:00:02:02:69');

                onosH += self.outputOnosNetcfgHost ("CR 21", "01:02:69", 42, 3);
                onosH += self.outputOnosNetcfgHost ("CR 22", "02:02:69", 44, 3);

                self.addLink (spine2, cr21, 1, 1);
                self.addLink (spine2, cr22, 2, 1);

                cr31 = self.addHost ('cr31', mac='00:00:00:01:03:69');
                cr32 = self.addHost ('cr32', mac='00:00:00:02:03:69');

                onosH += self.outputOnosNetcfgHost ("CR 31", "01:03:69", 72, 3);
                onosH += self.outputOnosNetcfgHost ("CR 32", "02:03:69", 74, 3);

                self.addLink (spine3, cr31, 1, 1);
                self.addLink (spine3, cr32, 2, 1);

                # R-CORD DSLAM
                rdslam1 = self.addSwitch ("rdslam1", dpid="21011");
                rdslam2 = self.addSwitch ("rdslam2", dpid="21012");
                rdslam3 = self.addSwitch ("rdslam3", dpid="21013");

                onosSW += self.outputOnosNetcfgSwitch ("DSLAM 1", "21011", 3, 9);
                onosSW += self.outputOnosNetcfgSwitch ("DSLAM 2", "21012", 7, 9);
                onosSW += self.outputOnosNetcfgSwitch ("DSLAM 3", "21013", 11, 9);

                self.addLink (rdslam1, leaf1, 1, 11);
                self.addLink (rdslam2, leaf1, 1, 12);
                self.addLink (rdslam3, leaf1, 1, 13);

                # R-CORD DSL MODEM / CPEs MAC 00:00:00:<host#>:<group#>:<leaf#>

                dsl11 = self.addHost ('dsl11', mac='00:00:00:01:01:01');
                dsl12 = self.addHost ('dsl12', mac='00:00:00:02:01:01');
                dsl13 = self.addHost ('dsl13', mac='00:00:00:03:01:01');

                onosH += self.outputOnosNetcfgHost ("dsl11", "01:01:01", 2, 11);
                onosH += self.outputOnosNetcfgHost ("dsl12", "02:01:01", 3, 11);
                onosH += self.outputOnosNetcfgHost ("dsl13", "03:01:01", 4, 11);

                self.addLink (dsl11, rdslam1, 1, 11);
                self.addLink (dsl12, rdslam1, 1, 12);
                self.addLink (dsl13, rdslam1, 1, 13);

                dsl21 = self.addHost ('dsl21', mac='00:00:00:01:02:01');
```

```
                dsl22 = self.addHost ('dsl22', mac='00:00:00:02:02:01');
                dsl23 = self.addHost ('dsl23', mac='00:00:00:03:02:01');

                onosH += self.outputOnosNetcfgHost ("dsl21", "01:02:01", 6, 11);
                onosH += self.outputOnosNetcfgHost ("dsl22", "02:02:01", 7, 11);
                onosH += self.outputOnosNetcfgHost ("dsl23", "03:02:01", 8, 11);

                self.addLink (dsl21, rdslam2, 1, 11);
                self.addLink (dsl22, rdslam2, 1, 12);
                self.addLink (dsl23, rdslam2, 1, 13);

                dsl31 = self.addHost ('dsl31', mac='00:00:00:01:03:01');
                dsl32 = self.addHost ('dsl32', mac='00:00:00:02:03:01');
                dsl33 = self.addHost ('dsl33', mac='00:00:00:03:03:01');

                onosH += self.outputOnosNetcfgHost ("dsl31", "01:03:01", 10, 11);
                onosH += self.outputOnosNetcfgHost ("dsl32", "02:03:01", 11, 11);
                onosH += self.outputOnosNetcfgHost ("dsl33", "03:03:01", 12, 11);

                self.addLink (dsl31, rdslam3, 1, 11);
                self.addLink (dsl32, rdslam3, 1, 12);
                self.addLink (dsl33, rdslam3, 1, 13);

                # R-CORD OLT
                rolt1 = self.addSwitch ("rolt1", dpid="21021");
                rolt2 = self.addSwitch ("rolt2", dpid="21022");
                rolt3 = self.addSwitch ("rolt3", dpid="21023");

                onosSW += self.outputOnosNetcfgSwitch ("OLT 1", "21021", 15, 9);
                onosSW += self.outputOnosNetcfgSwitch ("OLT 2", "21022", 19, 9);
                onosSW += self.outputOnosNetcfgSwitch ("OLT 3", "21023", 23, 9);

                self.addLink (rolt1, leaf2, 1, 11);
                self.addLink (rolt2, leaf2, 1, 12);
                self.addLink (rolt3, leaf2, 1, 13);

                # R-CORD ONT MODEM / CPEs
                ont11 = self.addHost ('ont11', mac='00:00:00:01:01:02');
                ont12 = self.addHost ('ont12', mac='00:00:00:02:01:02');
                ont13 = self.addHost ('ont13', mac='00:00:00:03:01:02');

                onosH += self.outputOnosNetcfgHost ("ont11", "01:01:02", 14, 11);
                onosH += self.outputOnosNetcfgHost ("ont12", "02:01:02", 15, 11);
                onosH += self.outputOnosNetcfgHost ("ont13", "03:01:02", 16, 11);

                self.addLink (ont11, rolt1, 1, 11);
                self.addLink (ont12, rolt1, 1, 12);
                self.addLink (ont13, rolt1, 1, 13);

                ont21 = self.addHost ('ont21', mac='00:00:00:01:02:02');
                ont22 = self.addHost ('ont22', mac='00:00:00:02:02:02');
                ont23 = self.addHost ('ont23', mac='00:00:00:03:02:02');

                onosH += self.outputOnosNetcfgHost ("ont21", "01:02:02", 18, 11);
                onosH += self.outputOnosNetcfgHost ("ont22", "02:02:02", 19, 11);
                onosH += self.outputOnosNetcfgHost ("ont23", "03:02:02", 20, 11);

                self.addLink (ont21, rolt2, 1, 11);
                self.addLink (ont22, rolt2, 1, 12);
                self.addLink (ont23, rolt2, 1, 13);

                ont31 = self.addHost ('ont31', mac='00:00:00:01:03:02');
                ont32 = self.addHost ('ont32', mac='00:00:00:02:03:02');
                ont33 = self.addHost ('ont33', mac='00:00:00:03:03:02');

                onosH += self.outputOnosNetcfgHost ("ont31", "01:03:02", 22, 11);
                onosH += self.outputOnosNetcfgHost ("ont32", "02:03:02", 23, 11);
                onosH += self.outputOnosNetcfgHost ("ont33", "03:03:02", 24, 11);

                self.addLink (ont31, rolt3, 1, 11);
                self.addLink (ont32, rolt3, 1, 12);
                self.addLink (ont33, rolt3, 1, 13);

                # R-CORD OLT/GFAST
                roltgf1 = self.addSwitch ("roltgf1", dpid="21031");
                roltgf2 = self.addSwitch ("roltgf2", dpid="21032");
```

```
                roltgf3 = self.addSwitch ("roltgf3", dpid="21033");

                onosSW += self.outputOnosNetcfgSwitch ("OLT GF 1", "21031", 31, 9);
                onosSW += self.outputOnosNetcfgSwitch ("OLT GF 2", "21032", 43, 9);
                onosSW += self.outputOnosNetcfgSwitch ("OLT GF 3", "21033", 55, 9);

                self.addLink (roltgf1, leaf3, 1, 11);
                self.addLink (roltgf2, leaf3, 1, 12);
                self.addLink (roltgf3, leaf3, 1, 13);

                # R-CORD GFAST
                rgf11 = self.addSwitch ("rgf11", dpid="31011");
                rgf12 = self.addSwitch ("rgf12", dpid="31012");
                rgf13 = self.addSwitch ("rgf13", dpid="31013");

                onosSW += self.outputOnosNetcfgSwitch ("GF 11", "31011", 27, 11);
                onosSW += self.outputOnosNetcfgSwitch ("GF 12", "31012", 31, 11);
                onosSW += self.outputOnosNetcfgSwitch ("GF 13", "31013", 35, 11);

                self.addLink (rgf11, roltgf1, 1, 11);
                self.addLink (rgf12, roltgf1, 1, 12);
                self.addLink (rgf13, roltgf1, 1, 13);


                # R-CORD DSL MODEM / CPEs
                dslgf111 = self.addHost ('dslgf111', mac="00:00:01:01:01:03");
                dslgf112 = self.addHost ('dslgf112', mac="00:00:02:01:01:03");
                dslgf113 = self.addHost ('dslgf113', mac="00:00:03:01:01:03");

                onosH += self.outputOnosNetcfgHost ("dslgf111", "01:01:01:03", 26, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf112", "02:01:01:03", 27, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf113", "03:01:01:03", 28, 13);

                self.addLink (dslgf111, rgf11, 1, 11);
                self.addLink (dslgf112, rgf11, 1, 12);
                self.addLink (dslgf113, rgf11, 1, 13);


                dslgf121 = self.addHost ('dslgf121', mac="00:00:01:02:01:03");
                dslgf122 = self.addHost ('dslgf122', mac="00:00:02:02:01:03");
                dslgf123 = self.addHost ('dslgf123', mac="00:00:03:02:01:03");

                onosH += self.outputOnosNetcfgHost ("dslgf121", "01:02:01:03", 30, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf122", "02:02:01:03", 31, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf123", "03:02:01:03", 32, 13);

                self.addLink (dslgf121, rgf12, 1, 11);
                self.addLink (dslgf122, rgf12, 1, 12);
                self.addLink (dslgf123, rgf12, 1, 13);


                dslgf131 = self.addHost ('dslgf131', mac="00:00:01:03:01:03");
                dslgf132 = self.addHost ('dslgf132', mac="00:00:02:03:01:03");
                dslgf133 = self.addHost ('dslgf133', mac="00:00:03:03:01:03");

                onosH += self.outputOnosNetcfgHost ("dslgf131", "01:03:01:03", 34, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf132", "02:03:01:03", 35, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf133", "03:03:01:03", 36, 13);

                self.addLink (dslgf131, rgf13, 1, 11);
                self.addLink (dslgf132, rgf13, 1, 12);
                self.addLink (dslgf133, rgf13, 1, 13);


                # R-CORD GFAST
                rgf21 = self.addSwitch ("rgf21", dpid="31021");
                rgf22 = self.addSwitch ("rgf22", dpid="31022");
                rgf23 = self.addSwitch ("rgf23", dpid="31023");

                onosSW += self.outputOnosNetcfgSwitch ("GF 21", "31021", 39, 11);
                onosSW += self.outputOnosNetcfgSwitch ("GF 22", "31022", 43, 11);
                onosSW += self.outputOnosNetcfgSwitch ("GF 23", "31023", 47, 11);

                self.addLink (rgf21, roltgf2, 1, 11);
                self.addLink (rgf22, roltgf2, 1, 12);
                self.addLink (rgf23, roltgf2, 1, 13);
```

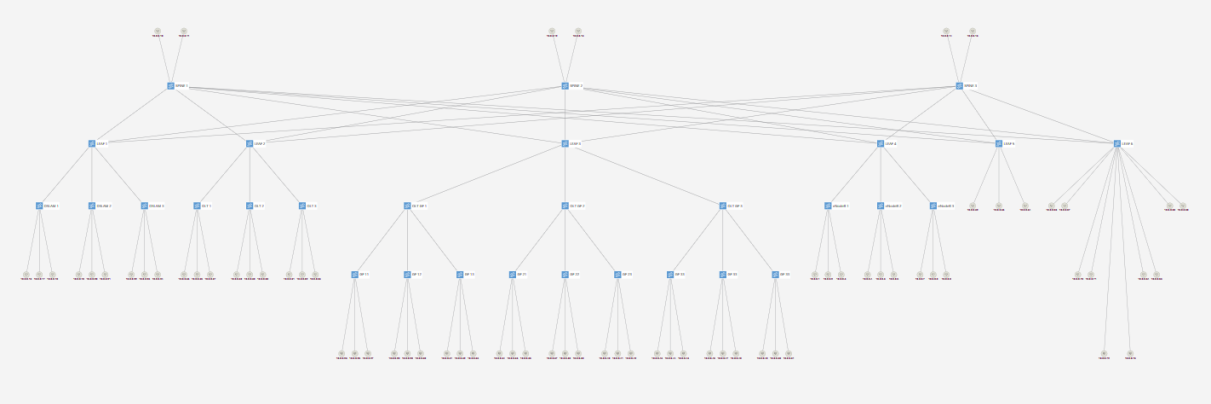
```
                # R-CORD DSL MODEM / CPEs
                dslgf211 = self.addHost ('dslgf211', mac="00:00:01:01:02:03");
                dslgf212 = self.addHost ('dslgf212', mac="00:00:02:01:02:03");
                dslgf213 = self.addHost ('dslgf213', mac="00:00:03:01:02:03");

                onosH += self.outputOnosNetcfgHost ("dslgf211", "01:01:02:03", 38, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf212", "02:01:02:03", 39, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf213", "03:01:02:03", 40, 13);

                self.addLink (dslgf211, rgf21, 1, 11);
                self.addLink (dslgf212, rgf21, 1, 12);
                self.addLink (dslgf213, rgf21, 1, 13);


                dslgf221 = self.addHost ('dslgf221', mac="00:00:01:02:02:03");
                dslgf222 = self.addHost ('dslgf222', mac="00:00:02:02:02:03");
                dslgf223 = self.addHost ('dslgf223', mac="00:00:03:02:02:03");

                onosH += self.outputOnosNetcfgHost ("dslgf221", "01:02:02:03", 42, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf222", "02:02:02:03", 43, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf223", "03:02:02:03", 44, 13);

                self.addLink (dslgf221, rgf22, 1, 11);
                self.addLink (dslgf222, rgf22, 1, 12);
                self.addLink (dslgf223, rgf22, 1, 13);


                dslgf231 = self.addHost ('dslgf231', mac="00:00:01:03:02:03");
                dslgf232 = self.addHost ('dslgf232', mac="00:00:02:03:02:03");
                dslgf233 = self.addHost ('dslgf233', mac="00:00:03:03:02:03");

                onosH += self.outputOnosNetcfgHost ("dslgf231", "01:03:02:03", 46, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf232", "02:03:02:03", 47, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf233", "03:03:02:03", 48, 13);

                self.addLink (dslgf231, rgf23, 1, 11);
                self.addLink (dslgf232, rgf23, 1, 12);
                self.addLink (dslgf233, rgf23, 1, 13);


                # R-CORD GFAST
                rgf31 = self.addSwitch ("rgf31", dpid="31031");
                rgf32 = self.addSwitch ("rgf32", dpid="31032");
                rgf33 = self.addSwitch ("rgf33", dpid="31033");

                onosSW += self.outputOnosNetcfgSwitch ("GF 33", "31031", 51, 11);
                onosSW += self.outputOnosNetcfgSwitch ("GF 33", "31032", 55, 11);
                onosSW += self.outputOnosNetcfgSwitch ("GF 33", "31033", 59, 11);

                self.addLink (rgf31, roltgf3, 1, 11);
                self.addLink (rgf32, roltgf3, 1, 12);
                self.addLink (rgf33, roltgf3, 1, 13);

                # R-CORD DSL MODEM / CPEs
                dslgf311 = self.addHost ('dslgf311', mac="00:00:01:01:03:03");
                dslgf312 = self.addHost ('dslgf312', mac="00:00:02:01:03:03");
                dslgf313 = self.addHost ('dslgf313', mac="00:00:03:01:03:03");

                onosH += self.outputOnosNetcfgHost ("dslgf311", "01:01:03:03", 50, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf312", "02:01:03:03", 51, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf313", "03:01:03:03", 52, 13);

                self.addLink (dslgf311, rgf31, 1, 11);
                self.addLink (dslgf312, rgf31, 1, 12);
                self.addLink (dslgf313, rgf31, 1, 13);

                dslgf321 = self.addHost ('dslgf321', mac="00:00:01:02:03:03");
                dslgf322 = self.addHost ('dslgf322', mac="00:00:02:02:03:03");
                dslgf323 = self.addHost ('dslgf323', mac="00:00:03:02:03:03");

                onosH += self.outputOnosNetcfgHost ("dslgf321", "01:02:03:03", 54, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf322", "02:02:03:03", 55, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf323", "03:02:03:03", 56, 13);

                self.addLink (dslgf321, rgf32, 1, 11);
```

```
                self.addLink (dslgf322, rgf32, 1, 12);
                self.addLink (dslgf323, rgf32, 1, 13);

                dslgf331 = self.addHost ('dslgf331', mac="00:00:01:03:03:03");
                dslgf332 = self.addHost ('dslgf332', mac="00:00:02:03:03:03");
                dslgf333 = self.addHost ('dslgf333', mac="00:00:03:03:03:03");

                onosH += self.outputOnosNetcfgHost ("dslgf331", "01:03:03:03", 58, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf332", "02:03:03:03", 59, 13);
                onosH += self.outputOnosNetcfgHost ("dslgf333", "03:03:03:03", 60, 13);

                self.addLink (dslgf331, rgf33, 1, 11);
                self.addLink (dslgf332, rgf33, 1, 12);
                self.addLink (dslgf333, rgf33, 1, 13);


                # M-CORD ENODEB
                menodeb1 = self.addSwitch ("menodeb1", dpid="41011");
                menodeb2 = self.addSwitch ("menodeb2", dpid="41012");
                menodeb3 = self.addSwitch ("menodeb3", dpid="41013");

                onosSW += self.outputOnosNetcfgSwitch ("eNodeB 1", "41011", 63, 9);
                onosSW += self.outputOnosNetcfgSwitch ("eNodeB 2", "41012", 67, 9);
                onosSW += self.outputOnosNetcfgSwitch ("eNodeB 3", "41013", 71, 9);

                self.addLink (menodeb1, leaf4, 1, 11);
                self.addLink (menodeb2, leaf4, 1, 12);
                self.addLink (menodeb3, leaf4, 1, 13);

                # M-CORD CELL PHONE
                cp11 = self.addHost ('cp11', mac="00:00:00:01:01:04");
                cp12 = self.addHost ('cp12', mac="00:00:00:02:01:04");
                cp13 = self.addHost ('cp13', mac="00:00:00:03:01:04");

                onosH += self.outputOnosNetcfgHost ("cp11", "01:01:04", 62, 11);
                onosH += self.outputOnosNetcfgHost ("cp12", "02:01:04", 63, 11);
                onosH += self.outputOnosNetcfgHost ("cp13", "03:01:04", 64, 11);

                self.addLink (cp11, menodeb1, 1, 11);
                self.addLink (cp12, menodeb1, 1, 12);
                self.addLink (cp13, menodeb1, 1, 13);

                cp21 = self.addHost ('cp21', mac="00:00:00:01:02:04");
                cp22 = self.addHost ('cp22', mac="00:00:00:02:02:04");
                cp23 = self.addHost ('cp23', mac="00:00:00:03:02:04");

                onosH += self.outputOnosNetcfgHost ("cp21", "01:02:04", 66, 11);
                onosH += self.outputOnosNetcfgHost ("cp22", "02:02:04", 67, 11);
                onosH += self.outputOnosNetcfgHost ("cp23", "03:02:04", 68, 11);

                self.addLink (cp21, menodeb2, 1, 11);
                self.addLink (cp22, menodeb2, 1, 12);
                self.addLink (cp23, menodeb2, 1, 13);

                cp31 = self.addHost ('cp31', mac="00:00:00:01:03:04");
                cp32 = self.addHost ('cp32', mac="00:00:00:02:03:04");
                cp33 = self.addHost ('cp33', mac="00:00:00:03:03:04");

                onosH += self.outputOnosNetcfgHost ("cp31", "01:03:04", 70, 11);
                onosH += self.outputOnosNetcfgHost ("cp32", "02:03:04", 71, 11);
                onosH += self.outputOnosNetcfgHost ("cp33", "03:03:04", 72, 11);

                self.addLink (cp31, menodeb3, 1, 11);
                self.addLink (cp32, menodeb3, 1, 12);
                self.addLink (cp33, menodeb3, 1, 13);

                # E-CORD
                ecpe11 = self.addHost ('ecpe11', mac="00:00:00:01:01:05");
                ecpe12 = self.addHost ('ecpe12', mac="00:00:00:02:01:05");
                ecpe13 = self.addHost ('ecpe13', mac="00:00:00:03:01:05");

                onosH += self.outputOnosNetcfgHost ("ecpe11", "01:01:05", 74, 9);
                onosH += self.outputOnosNetcfgHost ("ecpe12", "02:01:05", 76, 9);
                onosH += self.outputOnosNetcfgHost ("ecpe13", "03:01:05", 78, 9);

                self.addLink (ecpe11, leaf5, 1, 11);
```

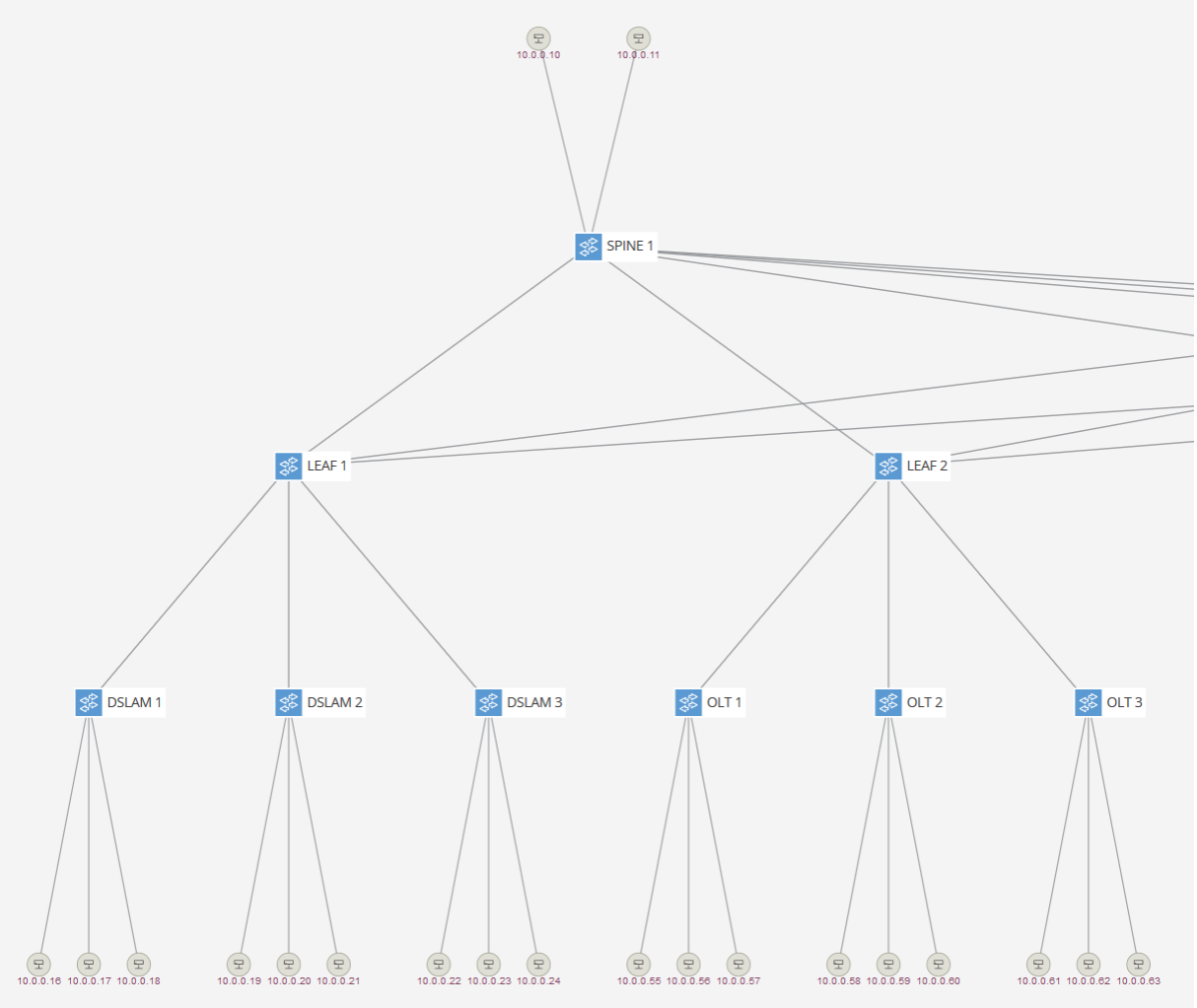
```
                self.addLink (ecpe12, leaf5, 1, 12);
                self.addLink (ecpe13, leaf5, 1, 13);


                # Service Edge
                sbng11 = self.addHost ('sbng11', mac="00:00:00:01:01:06");
                sbng21 = self.addHost ('sbng21', mac="00:00:00:02:01:06");

                onosH += self.outputOnosNetcfgHost ("sbng11", "01:01:06", 80, 9);
                onosH += self.outputOnosNetcfgHost ("sbng21", "02:01:06", 81, 9);

                self.addLink (sbng11, leaf6, 1, 11);
                self.addLink (sbng21, leaf6, 1, 12);

                sdhcp11 = self.addHost ('sdhcp11', mac="00:00:00:01:02:06");
                sdhcp21 = self.addHost ('sdhcp21', mac="00:00:00:02:02:06");

                onosH += self.outputOnosNetcfgHost ("sdhcp11", "01:02:06", 82, 11);
                onosH += self.outputOnosNetcfgHost ("sdhcp21", "02:02:06", 83, 11);

                self.addLink (sdhcp11, leaf6, 1, 13);
                self.addLink (sdhcp21, leaf6, 1, 14);

                sdns11 = self.addHost ('sdns11', mac="00:00:00:01:03:06");
                sdns21 = self.addHost ('sdns21', mac="00:00:00:02:03:06");

                onosH += self.outputOnosNetcfgHost ("sdns11", "01:03:06", 84, 13);
                onosH += self.outputOnosNetcfgHost ("sdns21", "02:03:06", 86, 13);

                self.addLink (sdns11, leaf6, 1, 15);
                self.addLink (sdns21, leaf6, 1, 16);

                saaa11 = self.addHost ('saaa11', mac="00:00:00:01:04:06");
                saaa21 = self.addHost ('saaa21', mac="00:00:00:02:04:06");

                onosH += self.outputOnosNetcfgHost ("saaa11", "01:04:06", 87, 11);
                onosH += self.outputOnosNetcfgHost ("saaa21", "02:04:06", 88, 11);

                self.addLink (saaa11, leaf6, 1, 17);
                self.addLink (saaa21, leaf6, 1, 18);

                scache11 = self.addHost ('scache11', mac="00:00:00:01:05:06");
                scache21 = self.addHost ('scache21', mac="00:00:00:02:05:06");

                onosH += self.outputOnosNetcfgHost ("scache11", "01:05:06", 89, 9);
                onosH += self.outputOnosNetcfgHost ("scache21", "02:05:06", 90, 9);

                self.addLink (scache11, leaf6, 1, 19);
                self.addLink (scache21, leaf6, 1, 20);


                print 'curl --user onos:rocks -X POST -H "Content-Type: application/json"
http://sdn-onos-dev-01:8181/onos/v1/network/configuration/ -d \'';
                print '{';
                print '"devices" : {';
                print onosSW;
                print '"of:0000000000000000" : { "basic": { "name":"NOT_EXISTENT" } }';
                print '},';
                print '"hosts" : {';
                print onosH;
                print '"00:00:00:00:00:00/-1" : { "basic": { "name":"NOT_EXISTENT" } }';
                print '}';
                print '}';
                print '\'';

topos = { 'WCORD': ( lambda: classWCORD() ) }
```

7.6 Finale Ansicht

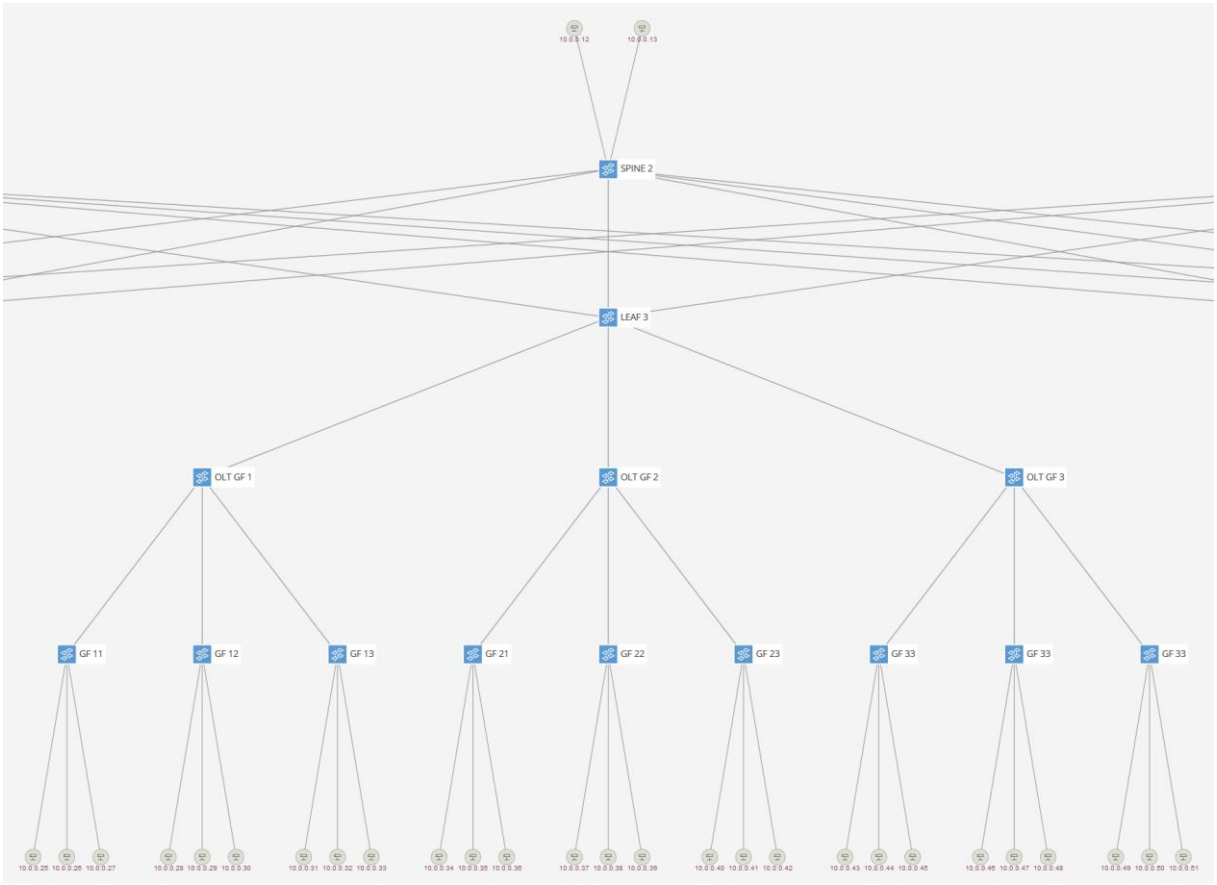
Das Gesamtbild



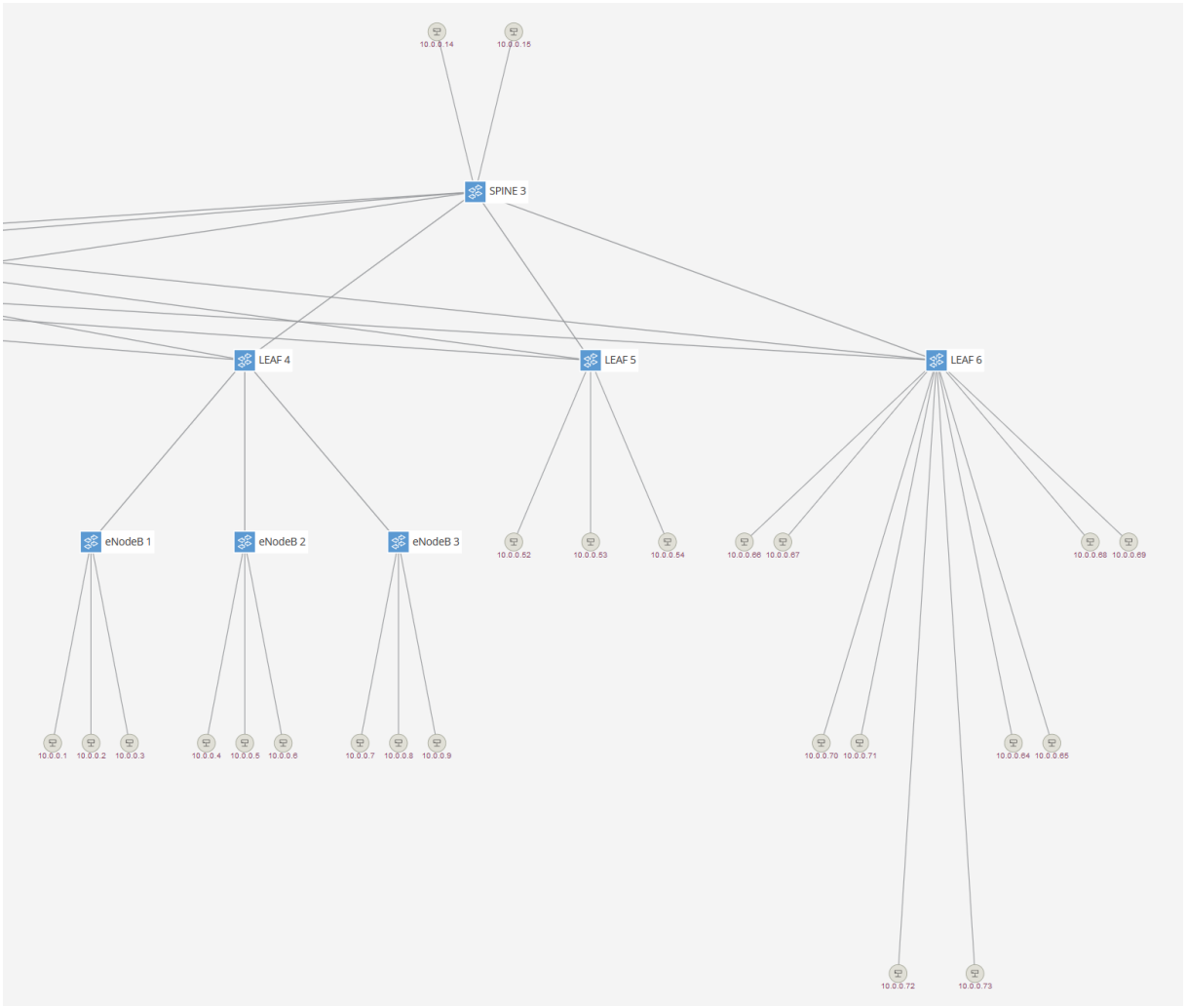
Der Bereich Leaf 1 und Leaf 2 mit Spine 1



Der Bereich Leaf 3 und Spine 2



Und der Leaf 4 und Leaf 5 und Leaf 6 mit Spine 3 und den Service Nodes


10.0.0.14
10.0.0.15
SPINE 3
LEAF 4
LEAF 5
LEAF 6
eNodeB 1
eNodeB 2
eNodeB 3
10.0.0.52
10.0.0.53
10.0.0.54
10.0.0.66
10.0.0.67
10.0.0.68
10.0.0.69
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
10.0.0.8
10.0.0.9
10.0.0.70
10.0.0.71
10.0.0.64
10.0.0.65
10.0.0.72
10.0.0.73