

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Uniwersalne Metody Projektowania Aplikacji na Urządzenia Mobilne i Wbudowane

Implementacja switch'a SDN na banana PI R2

skład sekcji:	Rafał Broncel Jan Nieora
kierunek:	Informatyka
grupa:	ISMIP1

1 Opis projektu

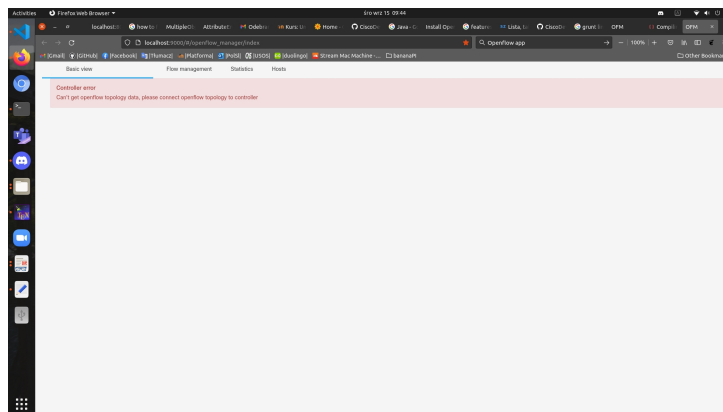
W celu zrealizowania switch'a SDN na banana PI R2 wykorzystaliśmy system operacyjny Armbian 21.03. Dodatkowo w roli kontrolera wykorzystaliśmy program Opendaylight w wersji karaf 08.04 wraz z zewnętrzną nakładką graficzną OpenflowApp. Kontroler był uruchomiony na komputerze z systemem Ubuntu 20.04. Dodatkowo w celu łatwiejszej konfiguracji oraz przepływu pakietów OpenFlow wykorzystaliśmy zew. kartę sieciową podłączoną do gniazda USB.

2 uruchomienie i konfiguracja kontrolera

W celu uruchomienia Opendaylight'a należy pobrać środowisko JAVA 8 JRE. Następnie wystarczy rozpakować skompresowany folder i uruchomić program poleceniem: `./karaf-0.8.4/bin/karaf`. Konieczne jest zainstalowanie trzech podstawowych features'ów:

- `odl-openflowplugin-app-topology-manager`
- `odl-l2switch-all`
- `odl-restconf-all`

W celu uruchomienia nakładki graficznej należy wejść do pobrango folderu z Open OpenflowApp i wykonać polecenie `grunt`. Następnie w przeglądarce uruchomić stronę: `http://localhost:9000`. Przeglądarka powinna pokazać stronę przedstawioną na rysunku 1.

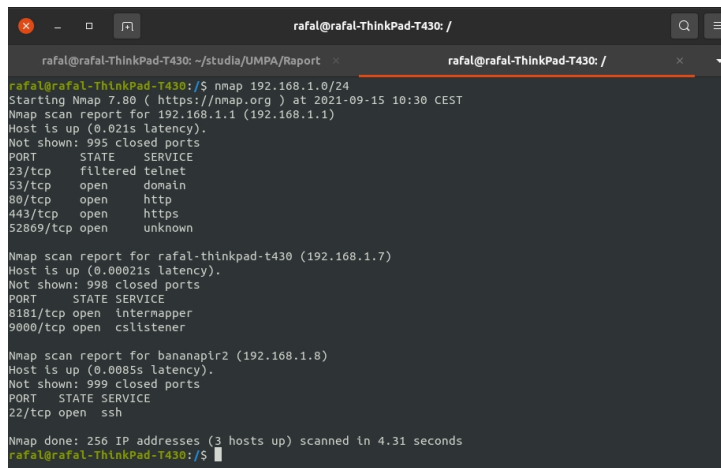


Rysunek 1: Uruchomienie nakładki graficznej bez uruchomionego banana PI R2

Po powyższych krokach ODL z OFM powinny być poprawnie skonfigurowane.

3 Uruchomienie i konfiguracja banana PI R2

Po zgraniu systemu na kartę sd należy podłączyć urządzenie do sieci w celu znalezienia banana pi'a wykorzystaliśmy polecenie `nmap` należy znać sieć i jej maskę. Rysunek 2 przedstawia wykonanie polecenia w domowej sieci



```
rafal@rafal-ThinkPad-T430: /
rafal@rafal-ThinkPad-T430: ~/studia/UMPA/Raport
rafal@rafal-ThinkPad-T430:/$ nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-15 10:30 CEST
Nmap scan report for 192.168.1.1 (192.168.1.1)
Host is up (0.021s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
23/tcp    filtered telnet
53/tcp    open  domain
80/tcp    open  http
443/tcp    open  https
52869/tcp  open  unknown

Nmap scan report for rafal-thinkpad-t430 (192.168.1.7)
Host is up (0.00021s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
8181/tcp  open  intermapper
9000/tcp  open  cslistener

Nmap scan report for bananapi2 (192.168.1.8)
Host is up (0.0085s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 256 IP addresses (3 hosts up) scanned in 4.31 seconds
rafal@rafal-ThinkPad-T430:/$
```

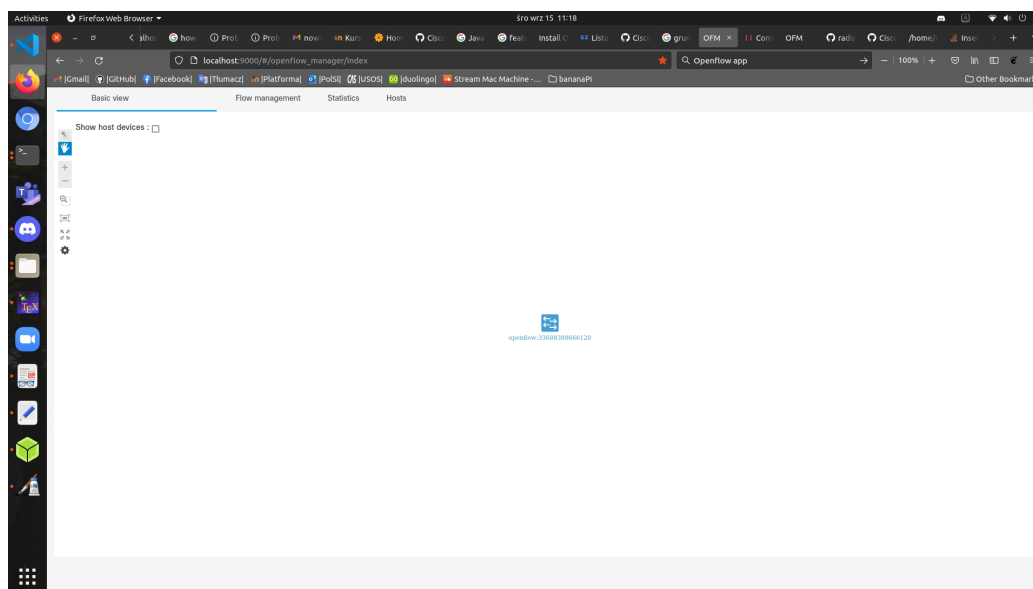
Rysunek 2: Uruchomienie polecenia nmap

W mojej sieci polecenie określa który adres ip należy do urządzenia, niestety w niektórych sieciach tak się nie dzieje więc trzeba zwracać uwagę na otwarte porty w przypadku banana pi'a otwarty jest tylko port 22. Aby się połączyć z urządzeniem należy wykonać polecenie: `ssh root@<adres ip banana pi>`. Po połączeniu możemy przystąpić do konfiguracji.

Pierwszym krokiem jest zainstalowanie `Open vSwitch'a`. Po instalacji musimy stworzyć most oraz ustawić adres ip kontrolera. , co jest pokazane poniżej:

- `ovs-vsctl add-br br1`
- `ovs-vsctl set-controller br1 tcp:<adres ip kontrolera>`

Po tych krokach ODL powinien wykryć banana pi'a jako switch'a. Przeglądarka z włączonym OFM na adresie `http://localhost:9000` powinna przedstawiać podobny widok tak jak rysunek 3



Rysunek 3: Uruchomienie banana PI z wstępnie skonfigurowanym swatchem

Kolejnym korkiem jest uruchomienie poniższego skryptu, który musi być uruchamiany za każdym razem jeśli banana pi nie zapamiętał ustawień, niestety nie zawsze zapamiętuje ustawienia.

```
ip link set dev wan down
ip link set dev eth0 down
ip link set dev wan address ca:fe:ca:fe:ca:fe
sudo ip link set dev eth0 address 12:12:12:12:12:12
ip link set dev wan up
```

```
#split eth0
```

```
ip link set eth0 up
ip link add link eth0 name eth0.10 type vlan id 10
ip link add link eth0 name eth0.20 type vlan id 20
ip link add link eth0 name eth0.30 type vlan id 30
ip link add link eth0 name eth0.40 type vlan id 40
ip link add link eth0 name eth0.50 type vlan id 50
```

```
ip link set dev eth0.10 down
ip link set dev eth0.20 down
ip link set dev eth0.30 down
ip link set dev eth0.40 down
ip link set dev eth0.50 down
```

```
#change mac
```

```
ip link set dev eth0.10 address 12:12:12:12:12:10
ip link set dev eth0.20 address 12:12:12:12:12:20
ip link set dev eth0.30 address 12:12:12:12:12:30
ip link set dev eth0.40 address 12:12:12:12:12:40
ip link set dev eth0.50 address 12:12:12:12:12:50
```

```
ip link set lan0 address ca:fe:ca:fe:ca:c0
ip link set lan1 address ca:fe:ca:fe:ca:c1
ip link set lan2 address ca:fe:ca:fe:ca:c2
ip link set lan3 address ca:fe:ca:fe:ca:c3
```

```
ip link set dev wan up
```

```
#remove
```

```
bridge vlan del vid 1 dev lan0
bridge vlan del vid 1 dev lan1
bridge vlan del vid 1 dev lan2
bridge vlan del vid 1 dev lan3
bridge vlan del vid 1 dev wan
```

```
#correct tag
```

```
bridge vlan add vid 10 dev lan0 pvid untagged
bridge vlan add vid 20 dev lan1 pvid untagged
bridge vlan add vid 30 dev lan2 pvid untagged
bridge vlan add vid 40 dev lan3 pvid untagged
bridge vlan add vid 50 dev wan pvid untagged
```

```
#turn on
```

```
ip link set dev br0 up
ip link set eth0.10 up
ip link set eth0.20 up
ip link set eth0.30 up
ip link set eth0.40 up
ip link set eth0.50 up
```

```
ip link set lan0 nomaster
ip link set lan1 nomaster
ip link set
\newpagelan2 nomaster
ip link set lan3 nomaster
ip link set wan nomaster
```

```
ip link set wan up
ip link set lan0 up
ip link set lan1 up
ip link set lan2 up
ip link set lan3 up
```

Teraz przechodzimy do konfiguracji samego **Open vSwitch**'a. Po wyonaniu skryptu możemy przejść do konfiguracji switcha. W tym celu musimy na początku wyłączyć lan'y i eth. W tym celu wykonujemy następujące komendy (które wystarczy uruchomić jednorazowo):

```
ip link set dev eth0 down
ip link set dev eth0.10 down
ip link set dev eth0.20 down
ip link set dev eth0.30 down
ip link set dev eth0.40 down
ip link set dev eth0.50 down
```

```
ip link set wan down
ip link set lan0 down
ip link set lan1 down
```

```
ip link set lan2 down
ip link set lan3 down
```

Kolejnym krokiem jest dodanie powyższych interfejsów do Open vSwitch'a więc trzeba wykonać następujące polecenia:

```
ovs-vsctl add-port br1 eth0.10
ovs-vsctl add-port br1 eth0.20
ovs-vsctl add-port br1 eth0.30
ovs-vsctl add-port br1 eth0.40
ovs-vsctl add-port br1 eth0.50

ovs-vsctl add-port br1 wan
ovs-vsctl add-port br1 lan0
ovs-vsctl add-port br1 lan1
ovs-vsctl add-port br1 lan2
ovs-vsctl add-port br1 lan3
```

Następnie musimy włączyć interfejsy:

```
ip link set dev eth0 up
ip link set dev eth0.10 up
ip link set dev eth0.20 up
ip link set dev eth0.30 up
ip link set dev eth0.40 up
ip link set dev eth0.50 up

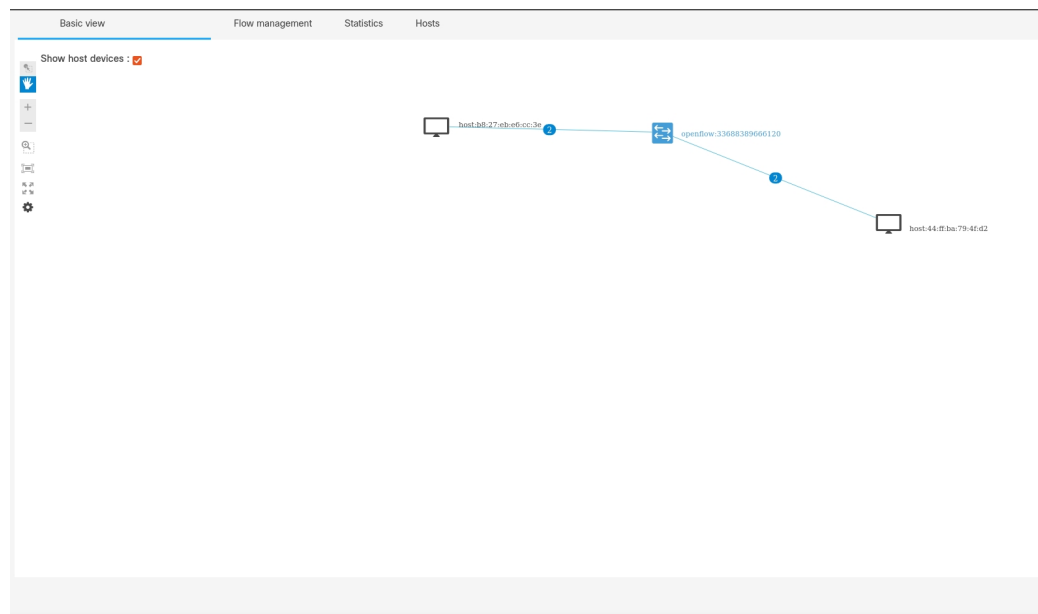
ip link set wan up
ip link set lan0 up
ip link set lan1 up
ip link set lan2 up
ip link set lan3 up
```

Po powyższych krokach możemy wykonać polecenie `ovs-vsctl show`, które powinno nam przedstawić podobny rezultat co rysunek 4. Tym samym powinniśmy mieć w pełni skonfigurowanego switcha. Istotnym faktem jest to, że switch wykryje tylko urządzenia jeśli przejdzie przez niego jakiś ruch sieciowy z urządzenia. Wystarczy użyć polecenia `ping` na dowolny adres ip. Kolejną ważnym szczegółem jest fakt, że jeśli będziemy blokować cały ruch to banana pi nie wykryje nowych urządzeń.


```
root@bananapir2:/etc/network# ovs-vsctl show
eb567cab-e933-4372-aba3-7fd36f0628bb
    Manager wan
    Bridge br1
        Controller "tcp:192.168.1.7"
            is_connected: true
        Port lan3
            Interface lan3
        Port eth0.50
            Interface eth0.50
        Port lan2
            Interface lan2
        Port wan
            Interface wan
        Port lan1
            Interface lan1
        Port br1
            Interface br1
                type: internal
        Port lan0
            Interface lan0
        Port eth0.20
            Interface eth0.20
        Port eth0.30
            Interface eth0.30
        Port eth0.10
            Interface eth0.10
        Port eth0.40
            Interface eth0.40
    ovs_version: "2.13.1"
root@bananapir2:/etc/network#
```

Rysunek 4: Skonfigurowany ovs

Rysunek 5 przedstawia główny widok OFM'a który świadczy o porawnej konfiguracji i otym że switch wykrył podłączone urządzenia.



Rysunek 5: Dev

4 Ogólny opis przebiegu projektu wraz z problemami napotkanymi

placeholder

5 Uwagi końcowe i wnioski

placeholder

Źródła

- Instalacja Javy wraz z uruchomieniem ODL
- OFM