25th International Scientific Conference UNITECH

INTERNATIONAL SCIENTIFIC CONFERENCE
20-22 November 2025
GABROVO

ИЗСЛЕДВАНЕ ВЪЗМОЖНОСТИТЕ НА ГРАФИЧНИ СРЕДИ ЗА УПРАВЛЕНИЕ НА КОНТРОЛЕР В SDN

Делян Генков, Мирослав Славов*

[2]Технически университет – Габрово, кат. „Компютърни системи и технологии“
ул. Хаджи Димитър 4, Габрово, България
**кореспондиращ автор: miroslav_slavov@tugab.bg*

STUDY THE CAPABILITIES OF GRAPHICAL ENVIRONMENTS FOR MANAGEMENT OF SDN CONTROLLER

Delyan Genkov, Miroslav Slavov*

Technical University of Gabrovo, department "Computer System and Technologies"
4 "Hadji Dimitar" str, Gabrovo, Bulgaria
**Corresponding author: miroslav_slavov@tugab.bg*

Abstract
The Software Defined Network (SDN) controller is the device which controls the network and provides the necessarily logic and intelligence in the network. Every controller provides interface for the administrator to deploy logic according to the needs. Usually this interface is command line (CLI) based or through script files. This require specific skills and knowledges for every single controller. Graphical User Interface (GUI) and more specific Web based GUI is one solution which could provide relatively unique way to configure different controllers.
Keywords: OpenDaylight controller; SDN; Web Interface.

ВЪВЕДЕНИЕ

Възможността да се достъпва и управлява контролера в софтуерно дефинираните мрежи е решаваща по отношение администрирането на мрежата и управлението на трафика в нея. Някои компании, лидери в областта на мрежовите технологии разработват собствени контролери, както и потребителски интерфейси за тяхното управление. Но тези разработки са защитени с лицензи и в повечето случаи не са свободни за употреба.

Отговор на тези разработки са контролерите, разработвани като отворен код. Недостатъка при тях е, че в повечето случаи тези контролери се управляват чрез специфични интерфейси, базирани на команден ред (Command Line Interface – CLI) или с помощта на скриптови файлове. Всичко това изисква специфични умения и познания от страна на администратора и възможност да се администрира само един контролер.

ИЗЛОЖЕНИЕ

SDN контролерите са отворен код биха могли да бъдат разглеждани като алтернатива на контролерите, разработвани от водещи компании в областта на мрежовото оборудване.

Сравнение на тези контролери е показано в таблица 1 [1].

Print ISSN: 1313-230X, Online ISSN: 3033-1404

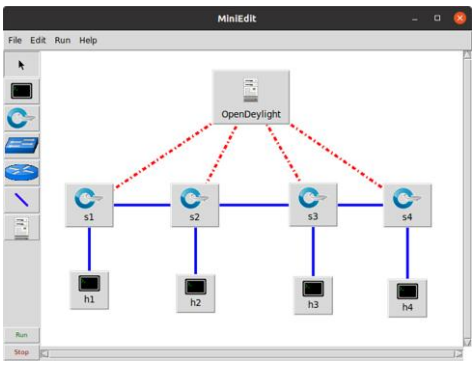
***Таблица 1**. Сравнение на контролери [1]*

	NOX	POX	Ryu	ODL
Език	C++	Python	Python	Java
Произв.	бърз	бавен	бавен	бърз
Разпределен	не	не	да	да
Open Flow	1.0	1.0	<=1.5	1.3
учене	средно	лесно	средно	трудно

В този доклад ще се фокусираме върху контролера с отворен код OpenDaylight. За този контролер има разработен уеб базиран графични интерфейс, който се предоставя заедно със самият него. Освен това някои водещи компании също са базирали свои разработки на този контролер. Тук ще се разгледа и уеб интерфейс, разработен от Cisco Systems – OpenDaylight OpenFlow Manager (OFM), който компанията, заедно с контролера, интегрира в свои по-мащабни разработки.

Описание на опитната постановка

OpenDaylight контролерът е инсталиран върху виртуална машина с операционна система Ubuntu. На същата виртуална машина е инсталиран и Cisco OFM.



***Фиг. 1**. Схема на реализираната мрежова топология*

За да се тестват възможностите на контролера и интерфейсите за управление, той ще бъде интегриран във виртуална среда на симулатор mininet. Mininet средата ще бъде стартирана на отделна виртуална машина, като ще бъдат създадена мрежа от няколко комутатора (Open vSwitch), които ще бъдат управлявани от OpenDaylight контролера.

Схема на топологията на мрежата е показана на фиг. 1.

Инсталиране на OpenDaylight

OpenDaylight контролерът е приложение, което се предоставя като контейнер. Може да бъде изтеглен в архив (.zip или .tar.gz). Тъй като контролерът е написан програмен език Java, предварително в системата трябва да има инсталирана правилната версия на Java работна среда – Java Runtime Environment (JRE). Средата е Apache Karaf.

Apache Karaf осигурява лека среда за изпълнение за инсталиране на допълнителни модули, които са включени в OpenDaylight платформата. По подразбиране OpenDaylight се инсталира без допълнителни модули [2].

След като желаната версия на контролера бъде изтеглена, той трябва да се разархивира в директория в системата. След това трябва да се стартира скрипта karaf в поддиректория bin (./bin/karaf). Това ще стартира контейнер, в които ще заради и контролерът. За стартирането на контейнерът е желателно да е настроена променлива на средата JAVA_HOME, въпреки, че контейнерът ще стартира и без нея (фиг. 2).

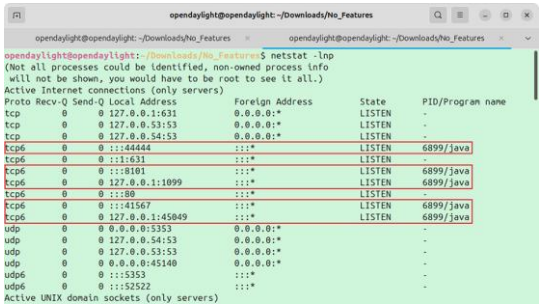
***Фиг. 2**. Стартирал karaf контейнер с предупреждение за липсваща глобална променлива JAVA_HOME*

След успешното стартиране на контейнерът с OpenDaylight, се стартира терминал, чрез който да се управлява контролерът и неговите модули (фиг. 3).

***Фиг. 3**. Стартиран терминал за управление на OpenSwitch*

Контролерът отваря и заема няколко порта в системата, но нито един от тях не може да се използва за комуникация с ко-

мутаторите (Open Flow протокол) или за управление на самият контролер (фиг. 4).



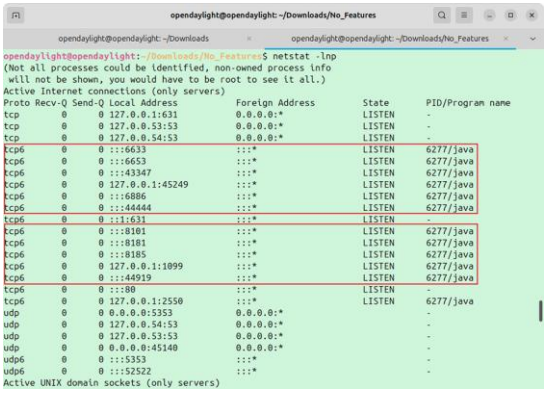
Фиг. 4*. Отворени от OpenDaylight портове преди да бъдат инсталирани модулите*

За целите на доклада, към контролерът ще бъдат инсталирани следните модули:

- *features-l2switch и features-openflowplugin* – чрез тези модули ще бъде инсталирано всичко, необходимо за да може контролерът да приема, обработва и изпраща съобщения, използвайки Open Flow протоколът и да управлява комутаторите.
- *features-mdsal* - Model-Driven Service Abstraction Layer (MD-SAL), предоставя инфраструктура за свързване на YANG модели към Java обектен модел и инфраструктура за предоставяне на YANG-дефинирани модели на взаимодействие[3][4].
- *features-restconf* – стартира всички модули, необходими на протокола RESCONF [5].
- *features-dlux и features-dluxapps* – инсталират се всички необходими модули за стартиране и поддръжка на Web графичния интерфейс, както и неговите компоненти.

Забележка: Модулите *l2switch,* както и модулите поддържащи графичната среда - *dlux и dluxapps* не са налични в най-новите версии на OpenDaylight.

След инсталирането на изброените модули броят на отворените и заети от контролерът комуникационни портове се увеличава (фиг. 5).



Фиг. 5*. Отворени от OpenDaylight портове след инсталирането на модулитете*

Изграждане на тестовата среда, стартиране и тестване на графичните интерфейси

За да се осигурят необходимите управляеми комутатори, които да се свържат към контролера, се използва симулаторът *mininet*. Използваните в средата комутатори са Open vSwitch (OVS), които са вградени в самият симулатор. Топологията на изградената и управлявана мрежа се състои от 4 комутатора, които контролерът ще управлява, свързани в линейна топология. Към всеки комутатор е свързан по 1 възел (host), чрез който ще се генерира тестов трафик (фиг. 1).

Командата за стартиране на тестовата мрежа е:

```
sudo mn --topo=linear,4 --mac --con-
trller=remote,ip=192.168.1.80,port=6653 -
-switch=ovs,protocols=OpenFlow13
```

където:

- *mn* – стартира mininet;
- *topo=linear,4* – дефинира топологията на мрежата, линейна с 4 комутатора;
- *mac* – дефинира опростен формат на MAC адресите на крайните хостове (напр. 00-00-00-00-00-01 и т.н.)
- *contrller=remote,ip=192.168.1.80, port=6653* – определя типа на контролера (локален или отдалечен), неговият адрес и номер на порт.

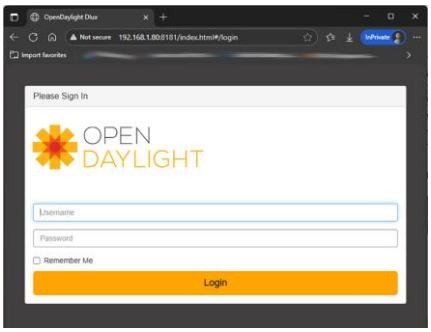
- *switch=ovs,protocols=OpenFlow13* – дефинира типа на използваните комутатори (Open vSwitch – OVS), както и протоколът за комуникация с контролера (Open Flow v 1.3).

При успешно стартиране и връзка с контролерът резултата трябва е като показаният на фиг. 6.

```
mininet@mininet-vm:~$ sudo mn --topo=linear,4 --mac --controller=remote,ip=192.168.1.80,port=6653 --switc
h=ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> _
```

***Фиг. 6**. Резултат от успешното стартиране на тестовата мрежа*

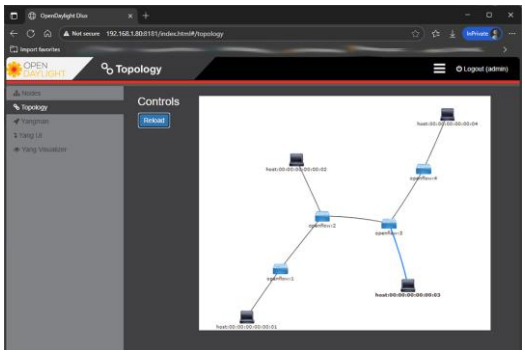
Сега се стартира вградения в уеб интерфейс. Достъпът до него става на адрес: *http://192.168.1.80:8181/index.html*



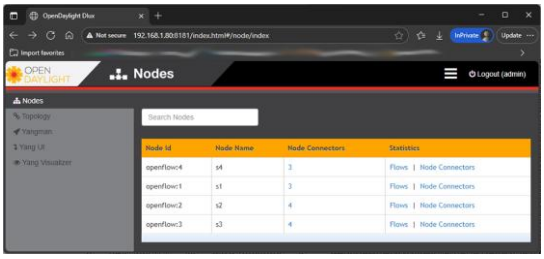
***Фиг. 7**. Страница за вход в уеб интерфейса*

След въвеждане на данни за вход (фиг. 7), може да се разгледа информацията, която контролерът има за мрежата – топология (фиг. 8), таблици с хостове (фиг. 9) и др.

За всеки комутатор може да се изведе информация за броя свързани портове, както и пакети/байтове, преминали през всеки интерфейс.

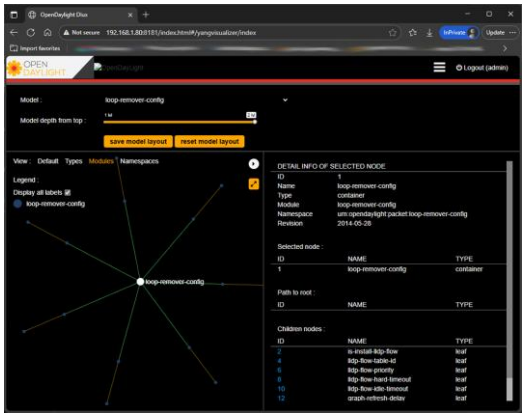


***Фиг. 8**. Топологията на тестовата мрежа наблюдавана от контролера*



***Фиг. 9**. Списък на комутаторите, управлявани от контролера*

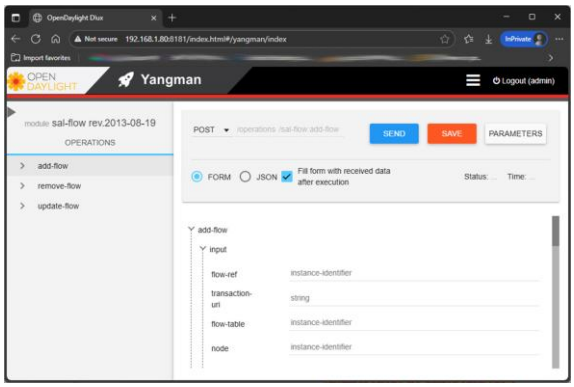
На фиг. 10 е показан един от YANG моделите, използвани от интерфейса.



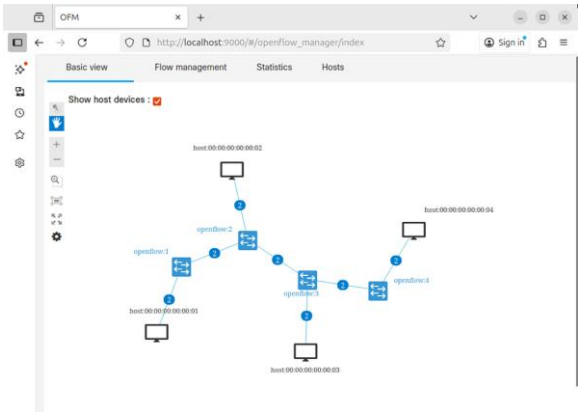
***Фиг. 10**. YANG модел на съобщенията за премахване на зацикляне (loop-remover-config)*

От тази страница може да се определи и модулът отговорен за създаването на съобщения в мрежата, напр. за моделът *add-flow* се използва модулът *sal-flow*, за добавяне, премахване и обновяване на потоци (фиг. 11)

Описанието на потокът става или чрез JSON формат или чрез попълване на полета от форма. Но описанието на полетата във формата не е съвсем ясно. Това изисква от администратора да притежава специфични познания и опит.



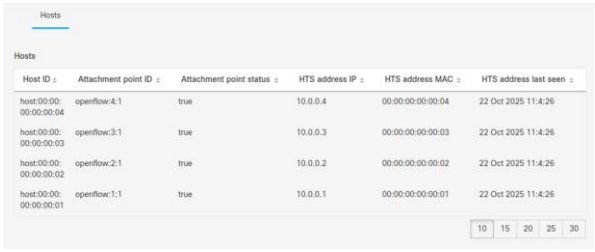
***Фиг. 11**. Добавяне/премахване/обновяване на потоци в контролера*



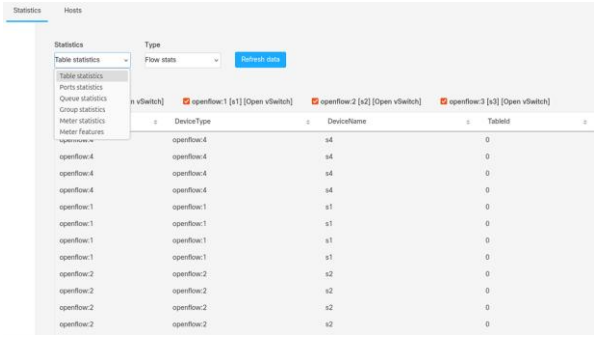
***Фиг. 12**. Топология на мрежата според OFM*

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> _
```

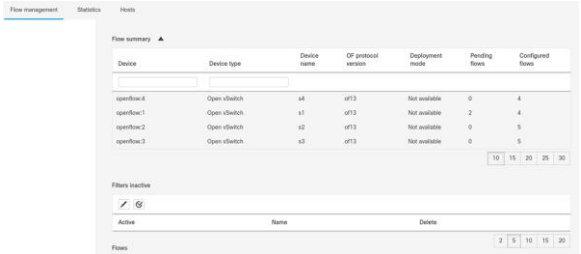
***Фиг. 13**. Тест на свързаността между всички възли в мрежата*



***Фиг. 14**. Информация за свързаните в мрежата хостове*



***Фиг. 15**. Статистическа информация от контролера*



***Фиг. 16**. Обобщена информация за потоците по устройства*

Cisco OpenDaylight OpenFlow Manager (OFM)

Cisco OFM е уеб интерфейс за OpenDaylight, разработен от Cisco DevNet, като към датата на изготвяне на доклада негова разширена версия е интегрирана в Cisco Open SDN Controller. OFM може да се изтегли от хранилището на DevNet в GitHub [6].

OFM също използва YANG модели за топологията и управлението на потоците. Освен това използва MD-SAL за да генерира набор от REST APIs (RESTCONF). Необходимите модули са: *odl-restconf-all; odl-openflowplugin-all* и *odl-l2switch-all* [6].

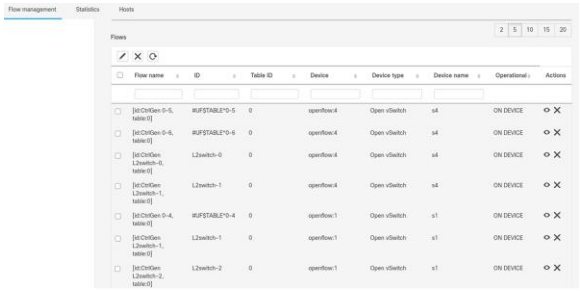
Достъпът до приложението става на адрес: *http://localhost:9000*.

Приложението дава по-оскъдна информация за мрежата, но тя е добре описана и възможните настройки и контроли са по-ясни. Приложението с топологията на мрежата (фиг. 12).

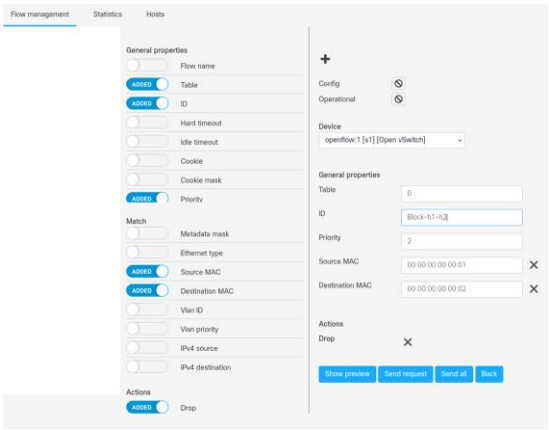
За да могат да се изобразят хостовете трябва да са генерирали трафик. За целта в mininet се изпълнява командата *pingall* (фиг. 13).

Най-важната част от интерфейса е статистиката за потоците (фиг. 16 и фиг. 17), както и възможността за добавяне на потоци в комутаторите (фиг. 18).

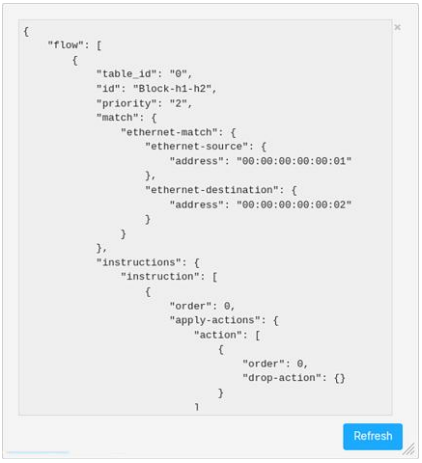
OFM интерфейса позволява да се прегледа информация за свързаните в мрежата хостове (фиг. 14), да се изведе статистика за различни показатели, като напр. таблиците с потоци (фиг. 15).



***Фиг. 17**. Подробна информация за потоците*



***Фиг. 18**. Добавяне на поток в таблицата на комутатор s1*

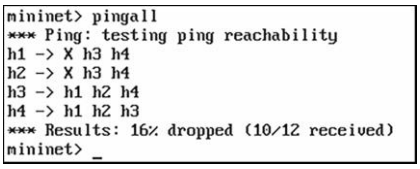


***Фиг. 19**. Добавеният поток в JSON формат*

Тук управлението на параметрите на потока е интуитивно и предполага възможност за управление от страна на администратори с по-малко опит.

След като се създаде потока, той може да се прегледа и като JSON обект, който ще се предаден към комутатора (фиг. 19).

След подаването на потока, комуникацията между *h1* и *h2* е блокирана (фиг. 20).



***Фиг. 20**. Блокиране на комуникацията след добавяне на поток от контролера*

ЗАКЛЮЧЕНИЕ

Разгледаните в доклада уеб базирани интерфейси имат своите положителни страни, но и в същото време имат и доста недостатъци, които правят използването им трудно. Целта на изследването е след като се разгледат налични решения, да се разработи алтернативен уеб базиран интерфейс за управление на SDN контролер, който да предостави възможности за работа дори и на хора, които нямат необходимите знания и опит.

Източник на финансиране: Настоящият документ е изготвен с финансовата помощ на договор № НИП2025-9 за провеждане на научни изследвания по проект на тема: „Изследване и изграждане на софтуерно дефинирани мрежи“ към Технически университет – Габрово.

ЛИТЕРАТУРА

[1] T. Raykov, Genkov, D., Slavov, M,., Experimental setup for testing of software defined networks controller, UNITECH’24, 21 – 22 November 2024, Gabrovo, pp 213-218.

[2] OpenDaylight concepts and tools - https://docs.opendaylight.org/en/latest/getting-started-guide/concepts_and_tools.html /посетен на 16.10.2025 г./.

[3] Model-Driven Service Abstraction Layer (MD-SAL) - https://docs.opendaylight.org/en/latest/release-notes/projects/mdsal.html /посетен на 16.10.2025 г./.

[4] YANG - https://en.wikipedia.org/wiki/YANG /посетен на 18.10.2025 г./.

[5] What is RESCONF - https://www.pynetlabs.com/what-is-restconf-protocol /посетен на 18.10.2025 г./.

[6] OpenDaylight OpenFlow Manager (OFM) App - https://github.com/CiscoDevNet/OpenDaylight-Openflow-App /посетен на 18.10.2025 г./.