Universitatea Politehnica București

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Structured Defined Networking

OpenFlow – VxLAN

Studenți: Profesor:

 Petrescu Theodor-Eduard Ștefan Stăncescu

Susanu Cristian Daniel

2015

Cuprins

[1. Software Defined Networking 3](#_Toc411455832)

[1.1 Introducere 3](#_Toc411455833)

[1.2 Arhitectura SDN 4](#_Toc411455834)

[1.3 Necesitatea SDN 6](#_Toc411455835)

[2. OpenFlow 7](#_Toc411455836)

[2.1 Descriere 7](#_Toc411455837)

[2.2 Operații 8](#_Toc411455838)

[2.3 Configurație 10](#_Toc411455839)

[3. Distribuire 11](#_Toc411455840)

[3.1 Sisteme distribuite 11](#_Toc411455841)

[4. Bibliografie 13](#_Toc411455842)

# Software Defined Networking

## 1.1 Introducere

Software Defined Networking (SDN) este un concept ce oferă operatorilor de rețea și centrelor de date flexibilitate în administrarea echipamentelor de rețea prin intermediul aplicațiilor software ce rulează pe servere externe. Acest concept propune renunțarea la dispunerea tradițională a stivei de rețea pentru a îmbunătății flexibilitatea și accesabilitatea. SDN permite customizării în masă a operațiilor rețelei pentru a suporta mai bine servicii preferențiale.

SDN este parte a unui grup de tehnologii ce deschid planurile de date, control și administrare a rețelei pentru a facilita accesul la structură prin intermediul API-urilor (Application Programming Interfaces).

Open Networking Foundation (ONF) este grupul cel mai implicat în crearea și standardizarea SDN, iar definiția oferită de aceștia este următoarea:

„SDN este o arhitectură în curs de dezvoltare, dinamică, ușor de administrat, eficientă din punct de vedere al costurilor și adaptabilă, făcând-o ideală pentru dimensiunea mare bandă necesară naturii dinamice a aplicațiilor din ziua de azi. Această arhitectură desprinde funcțiile de control de funcțiile de expediere, permițând astfel controlul rețelei să fie programabil direct și structura de sub acesta să fie abstractizată pentru aplicații și servicii. Protocolul OpenFlow este elementul fundamental în crearea de soluții SDN.”[1]

## 1.2 Arhitectura SDN

Arhitectura SDN este organizata pe 3 niveluri: aplicație, control și infrastructură.

 La nivelul de aplicație se află aplicațiile finite ce ajung la utilizatori, precum aplicații pentru video conferințe sau management-ul relațiilor cu clienții.

 Nivelul de control se referă la funcționalitatea ce permite aplicațiilor să execute eficient și în siguranță, precum firewall-uri și/sau protecția împotriva DDoS.

 Infrastructura este alcătuită din echipamentele în sine, aici regăsindu-se o gamă diferită de modalități de gestionare a acestora în funcție de necesitatea și scopul rețelei.

 Există astfel:
-Switch pur SDN - toate funcțiile de control ale unui switch tradițional (protocoalele de routare sunt folosite în crearea bazei de expediere a informației) sunt rulate de un controller central, astfel funcționalitatea switch-ului este restrânsă strict la planul de date.

-Switch-ul hibrid - este un switch pe care rulează, simultan, protocoalele de routare tradiționale și tehnologiile SDN). Un administrator de rețea poate configura controller-ul SDN pentru a descoperi și controla anumite fluxuri de trafic, în timp ce protocoalele tradiționale continuă să distribuie traficul prin rețea.

-Rețea hibridă – este rețeaua în care se regăsesc atât switch-uri tradiționale, pur SDN sau hibride funcționează în același sistem.



http://www.themetisfiles.com/wp-content/uploads/2012/10/software-defined-networking.png

Comunicarea între nivelul de control și infrastructură se realizează prin API-uri ce permit această comunicare, denumite Southbound API, iar aplicația standard pentru acest lucru este OpenFlow.

 Comunicarea între niveul aplicație și cel de control este realizat de Norhbound API, iar în momentul de față nu a fost standardizat un singur program pentru acest lucru.

 Controller-ul SDN reprezintă planul centralizat logic de control pentru toate componentele de rețea din raza domeniului său. Acesta distribuie și execută politici definite de administratorul de rețea într-un mod minuțios și ce acoperă un număr mare de device-uri. De asemenea, controller-ul introduce și abilitatea de programare în rețea, lucru considerat o premieră. Această programare permite folosirea librăriilor și a politicilor predefinite și asigură o platformă pe care se pot dezvolta, independent, noi tehnologii și aplicații de rețelistică pentru a satisface nevoile unei organizații.

 Controller-ul utilizează aplicația standard OpenFlow însă nu e exclusiv. Alte aplicații sunt NETCONF, OVSDB etc. Anumite controller-e au limitări de platformă ce permit utilizarea aplicațiilor/produselor unui singur producător.

## 1.3 Necesitatea SDN

Cel mai important aspect pentru care SDN este util este faptul că intrerupe liniaritatea în controlul rețelei și a componentelor de rețea permițând controlul acestora prin intermediul controller-ului SDN sau OpenFlow. Desigur acest lucru se poate anula dacă marii producători își vor produce versiunile proprii de controller.

Problemele a căror soluția o poate reprezenta tehnologia SDN sunt următoarele: utilizarea mai eficientă a resurselor rețelei, reducerea complexității de configurare, controlul traficului pe rețea cu o viziune clară aspura componentelor de intrare/ieșire, suport pentru mișcarea dinamică, replicarea și alocarea de resurse virtuale.

Producători ce se axează pe implementarea tehnologiei SDN sunt Cisco, HP, Huawei.

# OpenFlow

## 2.1 Descriere

OpenFlow este un protocol de comunicare standard. Acesta permite accesul la planul de expediere a unui switch sau router, facilitând un management mai sofisticat al traficului, în special pentru mediile de virtualizare și cloud. Acest protocol nu specifică un mecanism particular de tunelare, în schimb, permite extensii pentru realizarea oricărui protocol de tunelare dorit.

 Într-un router sau switch clasic, planul de date și cel de control se află pe aparat. OpenFlow permite diferitelor platforme din rețea, de diferite tipuri, să comunice cu un plan de control comun, numit controller aflat pe un server extern. Tradițional planul de control se ocupă doar de procesele de routare și direcționare.

 Avantajul real al utilizării OpenFlow constă în potențialele aplicații pe care le poate activa. Forme noi ale aplicațiilor tradiționale pentru sercuritate și, chiar aplicații noi, ce pot fi realizate cu ajutorul acestor controller-e.

Deciziile planului de control pentru fluxuri sunt stabilite de controller, iar celelalte decizii și controlul datelor este, în continuare, decis de router.

## 2.2 Operații

Tabelul de fluxuri conține date ce definesc un flux pe baza header-ului de pachet, și sunt sortate pe baza priorității definite de controller.

De la apariția sa, în decembrie 2009, OpenFlow v1.0 conținea un singur tabel de fluxuri și era axat pe identificarea de adrese Ipv4

 Versiunea v1.1 aducea îmbunătățiri la nivelul tabelelor, fiind capabil să susțină mai multe astfel de tabele, suport pentru VPLS ( o clasă de VPN ce suportă conexiunile a mai multor site-uri pe un singur domeniu administrat de o rețea IP/MPLS) și MPLS (Multiprotocol Label Switching) – un mecanism ce direcționează date de la un nod de rețea la altul bazat pe calea cea mai scurtă, pentru a evita căutările complexe în tabela de routare. De asemenea, a mai fost adăugat strategia ECMP (Equal-cost multi-path routing), o strategie în care expedierea pachetului dintr-un nod către o singură destinație se poate realiza pe mai multe „căi bune” care sunt la egalitate în topul costurilor. Routarea multipath poate fi utilizată în simultan cu majoritatea protocoalelor de routare, deoarece are decizie la nivel de nod și e limitată la un singur router.

Versiunea următoare, v1.2 a primit suport pentru IPv6, identificare după TLV (type-length-value) și mai multe controller-e.

În versiunea 1.3, s-au adăugat multiple canale paralele între switch și controller dar și IEEE 802.1 ah PBB, un set de protocoale și arhitecturi de routare pe rețeaua furnizorului ce permite interconectarea mai multor rețele fără a pierde individualitatea VLAN-urilor.

Pipeline-ul de rulare a unui pachet într-un switch OpenFlow



http://www.cisco.com/web/solutions/trends/open\_network\_environment/docs/cisco\_one\_webcastan\_introduction\_to\_openflowfebruary142013.pdf

## 2.3 Configurație

 Protocolul de configurație la OpenFlow se ocupa la început doar de configurare și management-ul traficului prin switch. Apoi au fost adăugate funcții ca: descoperirea capabilității, configurare de tunel, managementul erorilor și repararea unor mici inconsistențe.

 Se urmărește introducerea posibilității de a adăuga resurse switch-urilor logice, detecție simplă de topologie și notificări.



<http://www.cisco.com/web/solutions/trends/open_network_environment/docs/cisco_one_webcastan_introduction_to_openflowfebruary142013.pdf>

 OpenFlow este ideal în medii unde ser urmărește controlul precis al comportamentului rețelei.

 Cazuri de utilizare a protocolului OpenFlow:

1. Beneficiul primar al acestui protocol este natura centralizată a bazei de expediere a informației (FIB – Forwarding Information Base). Centralizarea permite calcularea rutelor optime pentru fiecare flux punând la dispoziție un model complet al topologiei rețelei. Acest model poate fi construit cu ajutorul unui protocol de descoperire (LLDP – Link Layer Discovery Protocol). Abilitatea de a controla traficul de la un capăt la altul al topologiei este o caracteristică ce îi permite asocierea cu SDN.
2. Optimizare pentru WAN (Wide Area Network) – traficul pe WAN poate fi deviat dinamic pentru a reduce latența pentru VoIP și alte aplicații sensibile din acest punct de vedere. Traficul poate fi controlat pe căi paralele de diferite costuri.
3. Virtualizare pe baza OpenFlow – o dată cu versiunea 1.3 a acestui protocol, un switch OpenFlow poate fi programat să realizeze încapsulare și decapsulare. Prin urmare, acest switch poate fi programat să fie un gateway sau un nivel de virtualizare VTEP/NVE.

# Distribuire

## Sisteme distribuite

Un sistem distribuit este alcătuit din mai multe computere autonome, conectate printr-o rețea, ce permite coordonarea activităților și împărțirea de resurse de sistem pentr ca user-ii să perceapă această multitudine de calculatoare ca un singur sistem integrat.

„O colecție de computer-e independente ce sunt prezentate user-ului ca un singur sistem coerent” – A. Tanenbaum.

„Partea hard și soft a unei colecții de computere independete ce cooperează pentru a realiza ceva funcțional.” – J. Lukkien

Un sistem distribuit este descris de anumite caracteristici, cum ar fi faptul că software-ul rulează în mai multe procese pe diferite procesoare, componentele nu sunt împărtășite de toți user-ii, mai multe puncte de control, mai multe puncte de eroare.

Un punct forte al unui sistem distribuit este scalabilitatea – abilitatea unui sistem, rețea să fie redimensionat pentru a satisface activități intense. Redimensionarea poate însemna adăugarea de noduri (rețea) sau adăugarea de piese pentru a crește puterea de calcul și performanțele.

Scalabilitatea este de 3 feluri:

 -în mărime: număr mare de computere, useri, procese

 -locație: se are de a face cu distribuția geografică a centrelor de date

 -administrație: costurile sunt supervizate de comitetul de administrație

De asemenea, se introduc și noi proprietăți ale acestor sisteme:

 -concurența

 -replicarea

 -securitatea

 -modularitatea

Concurența reprezintă abilitatea sistemului distribuit de a executa mai multe procese odată fiind implicate mai multe procesoare. Se poate ajunge la erori dacă update-urile componentelor concurente nu sunt coordonate.

Transparența sistemelor distribuite poate fi de mai multe feluri:

1. Transparența la acces: permite accesul la informația locală și cea remote să fie accesată folosind aceleași operații
2. Transparența locației: permite accesul la informație fără a se cunoaște locația acesteia
3. Transparența la concurență: permite proceselor să ruleze concurent folosind informație comună fără a interfera cu acestea.
4. Transparența la replicare: permite folosirea mai multor instanțe de informție pentru a crește fiabilitatea și performanța fără a semnala acest lucru user-ilor sau aplicațiilor
5. Transparența la erori: permite user-ilor și aplicațiilor să termine procesele cu toate că unele componente cedează
6. Transparența la migrare: permite transportul informației în sistem fără a afecta operațiile user-ilor sau a aplicațiilor
7. Transparența performanței: permite sistemului să fie reconfigurat pentru a îmbunătății performanța

Replicarea reprezintă procesul de a face copii datelor pe mai multe servere, permițând mai multor servere să fie implicate în compilare.

# Bibliografie

[1]<https://www.opennetworking.org/sdn-resources/sdn-definition>

[2]<http://www.cisco.com/web/solutions/trends/open_network_environment/docs/cisco_one_webcastan_introduction_to_openflowfebruary142013.pdf>

3. Software Defined Networking and OpenFlow – Eric Choi

4. Introduction to Software Definde Networking – OpenFlow & VxLAN – Vishal Shukla

5. <http://www.win.tue.nl/~johanl/educ/2II45/ADS.02.Distributed.pdf>

6. http://book.mixu.net/distsys/single-page.html