

## TENDINȚE DE EVOLUȚIE A COMUNICAȚIILOR SPRE INTERNETUL VIITORULUI

În ziua de azi, lumea comunicațiilor prin Internet cunoaște o efervescență tot mai crescută, cu un număr din ce în ce mai variat de servicii ce vin în întâmpinarea nevoilor utilizatorilor. În spatele acestor servicii recent apărute se află de cele mai multe ori tehnologii noi ce oferă posibilitatea schimbului de date la viteze din ce în ce mai mari, noi paradigme de accesare a informației și nu în ultimul rând noi metode de utilizare/partajare a resurselor din infrastructurile de comunicații existente.

În cadrul acestui context de cristalizare de noi direcții și tehnologii, apare conceptul de separare a planului de control al rutării de cel al comutației fluxurilor de date. Dacă în abordarea clasică ambele sunt implementate la nivelul echipamentelor de rutare, există în prezent o tendință clară de transfer a mecanismelor de control către un nivel superior, cât mai aproape de aplicații. Această abordare este cunoscută în literatura de specialitate sub numele de SDN (Software-Defined Networking – rețele definite prin *software*). Planul de control din paradigma SDN are acces direct la planul de comutație a datelor, ce înglobează echipamentele din infrastructura de rețea (ex. *router*-e, *switch*-uri și alte echipamente intermediare). Întregul efort de schimbarea a paradigmei rețelelor cu comutație de pachete are ca țel final facilitarea implementării de soluții inovative care să aducă îmbunătățiri majore comunicațiilor prin Internet, în cel mai scurt timp. Printre cel mai des întâlnite aplicații care au adoptat această abordare, putem aminti: controlul dinamic al accesului utilizatorilor în rețea, distribuirea eficientă a încărcării serverelor, virtualizarea resurselor rețelei, eficientizarea consumului de energie a rețelei,

migrarea automată a mașinilor virtuale fără întreruperea serviciilor [1].

Primele implementari comerciale de succes ce au adus tehnologia SDN în lumina reflectoarelor au fost: sistemul de management al traficului în WAN (Wide Area Network), dezvoltat de Google [2] și platforma de virtualizare a resurselor din rețea dezvoltată de Nicira, companie achiziționată de către VMware. În prezent, majoritatea marilor jucători din domeniul tehnologiei informației precum furnizori de servicii de tip *cloud*, furnizori de Internet, producători de echipamente și companii de servicii financiare au format diverse consorții în cadrul cărora își concentrează eforturile în vederea dezvoltării tehnologiei SDN : Open Networking Foundation [4], Open Daylight [5]). Astfel, putem afirma faptul că în prezent se fac pași concreți către realizarea unui vis demult enunțat și anume crearea rețelelor de date cu adevărat programabile care pot să se adapteze fără probleme la dinamica serviciilor disponibile și accesate în ziua de azi prin Internet. Din punct de vedere istoric se evidențiază trei faze ale evoluției spre ceea ce înseamnă în ziua de azi rețele programabile:

1. Din a doua jumătate a anilor `90 și până la începutul anilor 2000 a apărut conceptul de rețea activă ce a introdus funcții programabile în arhitectura rețelelor de calculatoare;
2. În perioada 2001 – 2007 s-a vehiculat ideea separării planului de comutație a datelor de cel de control datorită apariției de interfețe specializate ce au facilitat comunicația dintre cele două;
3. Începând cu anul 2007 apare tehnologia OpenFlow împreună cu diverse versiuni de sisteme de operare deschise, create special pentru echipamente de rețea.

Una dintre aplicațiile tehnologiei SDN considerate a fi cele mai importante la ora actuală și pentru dezvoltarea căreia se alocă în prezent cele mai multe resurse, este platforma de virtualizare a resurselor rețelei. Această fază este următorul pas logic ce este necesar a fi efectuat după apariția soluțiilor de virtualizare a puterii de calcul și a capacității de stocare. Un alt factor important ce accelerează dezvoltarea acestei direcții este apariția pe piață a operatorilor virtuali de telecomunicații. Aceștia utilizează parțial resursele infrastructurilor de comunicații a unuia sau mai multor furnizori, pentru a oferi servicii noi utilizatorilor.

Restul articolului este organizat după cum urmează: capitolul doi demonstrează că tehnologia OpenFlow este cea mai semnificativă din grupul de soluții SDN existente la ora actuală, urmând ca în capitolul trei să prezentăm

o aplicație în care aceasta este folosită cu succes pentru a optimiza *rutarea* datelor în interiorul rețelei. Capitolul patru conține rezultatele experimentale care confirmă avantajele protocolului de *rutare* propus. Capitolul cinci încheie această lucrare cu un set de concluzii ce sintetizează principalele idei ce trebuie reținute vizavi de conceptele prezentate.

## TEHNOLOGIA OPENFLOW

Mecanismele de tip SDN fac parte dintr-o arhitectură în care componenta de control a rețelei este decuplată de cea de comutație a datelor și adusă mai aproape de stratul aplicație prin biblioteci de funcții specializate (API – Application Programming Interface). În consecință, se permite optimizarea deciziilor luate în rețea astfel încât să se țină cont de cerințele serviciilor a căror date sunt transmise prin infrastructura de comunicații (vezi *Figura 1*). Un alt avantaj al acestei abordări este faptul că echipamentele din rețea pot fi mult simplificate deoarece acestea nu mai trebuie să implementeze o gamă foarte largă de protocoale deoarece toate deciziile sunt luate de către entitatea de control a rețelei. Singura cerință pentru aceste echipamente este ca ele să accepte și să execute instrucțiuni provenite de la entitatea de control. Astfel, configurarea securizarea și optimizarea infrastructurii se poate face automat cu programe specializate în executarea acestor funcții.

Prima soluție apărută ce permite interfațarea planului de comutație a datelor cu cel de control, în vederea implementării paradigmei SDN, este tehnologia OpenFlow. Aceasta permite accesul direct și controlul echipamentelor de comutație existente în infrastructura de rețea: comutatoare și *router-e* [6].

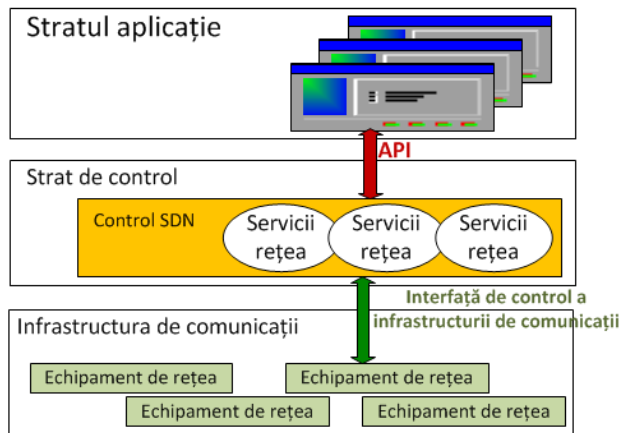


Figura 1. Arhitectura SDN

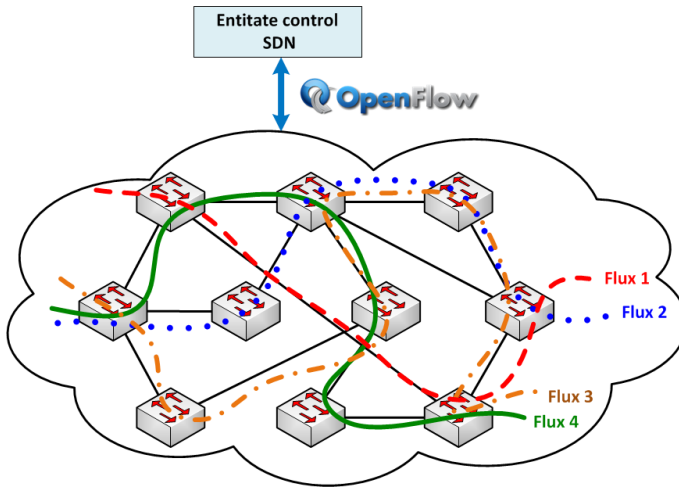


Figura 2. Control la nivel de flux cu ajutorul OpenFlow

În ceea ce privește granularitatea de reprezentare a datelor, se folosește noțiunea de flux ce identifică *unic* date care respectă un set predefinit de reguli. Spre deosebire de *rutarea* clasică, programarea funcționării rețelei la nivel de flux de date oferă un control fin ce permite infrastructurii să răspundă în timp real la toate schimbările ce apar datorită dinamicității aplicațiilor (vezi Figura 2). Cheia acestei abordări este tabela de fluxuri ce conține informații referitoare la setul de date ce străbat fiecare echipament în parte [7]. Pe lângă partea de identificare, o astfel de tabelă conține și acțiunea ce trebuie luată atunci când un pachet recepționat este detectat ca făcând parte dintr-un anumit flux de date. Această acțiune specifică de cele mai multe ori interfața de ieșire pe care trebuie redirectat mai departe pachetul în calea sa spre destinație (vezi Figura 3).

Utilizat inițial doar în infrastructuri bazate pe tehnologii din familia Ethernet, OpenFlow poate fi în ziua de azi implementat atât în rețele fizice ce utilizează o gamă largă de tehnologii cât și în rețele virtuale. Deoarece echipamentele ce suportă rutarea bazată pe mecanisme OpenFlow implementează și protocoale clasice de rutare, adoptarea acestei tehnologii SDN de către furnizorii de servicii se poate face progresiv, menținând costurile la nivele acceptabile.

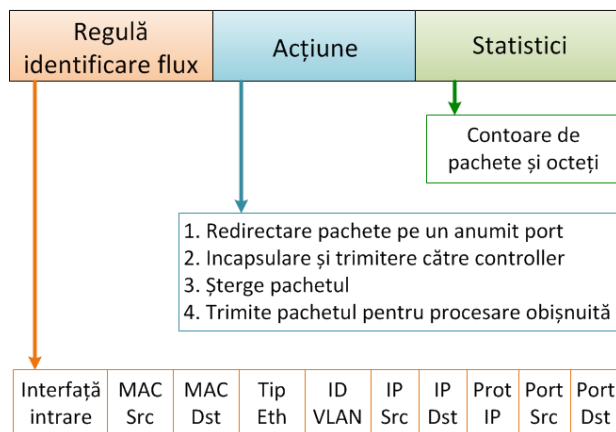


Figura 3. Tabela de fluxuri, existentă în echipamente compatibile OpenFlow

Odată cu apariția tehnologiei OpenFlow au fost elaborate o serie de soluții de control a rețelei ce se bazează pe aceasta: NOX, Beacon, Helios, Big Network Controller, SNAC și Maestro. Primul mecanism apărut ce se încadrează în această categorie a fost **NOX**. Acesta este de fapt un sistem de operare cu rol de control a rețelelor ce conțin echipamente de comutație OpenFlow, oferind suport pentru aplicații scrise în limbajele C++ și Python. O a doua soluție disponibilă pentru control este Beacon. Aceasta are ca principală caracteristică posibilitatea de a permite extinderea funcționalității sale prin adăugarea de module scrise în limbajul Java. De asemenea, are la bază *framework*-ul OSGi (Open Service Gateway initiative) ce permite aplicațiilor de control să fie pornite, oprite sau reinițializate fără a pierde conexiunea cu infrastructura controlată. Destinat îndeosebi grupurilor de cercetare, Helios este o soluție de gestiune a funcționalității rețelei elaborată de către NEC în limbajul de programare C. Aceasta oferă un interpretor de comenzi (en. *shell*) programabil destinat executării de experimente automatizate cu un grad ridicat de complexitate. Big Network Controller este un utilitar proprietar bazat pe Beacon și elaborat de către compania BigSwitch. Acesta oferă o interfață în linie de comandă destinată controlului rețelei. Bazată pe NOX, soluția SNAC face parte din categoria programelor publicate sub licență GNU (GNU's Not Unix!). Acesta oferă o interfață intuitivă și ușor de utilizat pentru controlul punctelor de comutație din rețelele de calculatoare folosind limbajul FML (Formal Modelling Language) cu ajutorul căruia se pot specifica politici de funcționare. Utilitarul Maestro elaborat de

către Rice University ca entitate de orchestrare a funcționării rețelei, este bazat pe limbajul Java și oferă posibilitatea de extensie a acestuia prin module adiționale adăugate în funcție de necesități. O altă caracteristică este faptul că a fost optimizat pentru rularea în paralel a mai multor fire de execuție în vederea creșterii performanțelor de control.

Printre beneficiile ce pot fi dobândite în urma utilizării tehnologiei OpenFlow am putea enumera [6]:

1. Creșterea securității rețelelor prin definirea la nivel înalt a politicilor corespunzătoare. Prin intermediul tehnologiei OpenFlow, acestea sunt translate în instrucțiuni specifice fiecărui echipament din rețeaua controlată. În cazul în care apar schimbări ale infrastructurii operaționale, actualizările se vor executa într-un timp mult mai scurt decât în abordarea clasică. Deoarece tehnica SDN oferă o vizibilitate globală asupra rețelei, se asigură faptul că setările de control acces, inginerie a traficului și calitate a serviciilor sunt implementate unitar în întreaga infrastructură, inclusiv campusuri, centre de date, filiale distante.
2. Controlul centralizat al unei infrastructuri de rețea multi-vendor în vederea punerii în funcțiune și a configurării cât mai rapide a acesteia.
3. Controlul bazat pe flux permite o gestiune granulară a întregii rețele astfel încât se pot specifica politici la nivel de utilizator, echipament sau aplicație. Acest fapt permite furnizorilor de servicii *cloud* să ofere suport pentru izolarea traficului, securitate și management elastic al resurselor în scenarii în care mai mulți utilizatori partajează aceeași infrastructură.
4. Creșterea satisfacției utilizatorilor. Acest beneficiu este datorat faptului că parametrii ce reflectă în timp real proprietățile rețelei sunt puși la dispoziția aplicațiilor de nivel înalt, astfel încât deciziile acestora să fie optimizate la situația actuală. Un exemplu concludent în acest sens ar putea fi eficientizarea serviciilor de tip IPTV (Internet Protocol Television) care transmit fluxuri video în timp real. Cunoscând starea rețelei și rata de transfer disponibilă, acestea pot modifica dinamic rata la care se face compresia fluxului astfel încât să adapteze automat cantitatea de date transmise pe canal.
5. Accelerarea inovației prin implementarea unor noi modele de business. De exemplu, oferirea de servicii de tipul *IT-as-a Service* în care este necesară posibilitatea modificării dinamice a modului de

funcționare a infrastructurii astfel încât să fie îndeplinite nevoile aplicațiilor și ale utilizatorilor.

6. Reducerea complexității de operare prin automatizarea proceselor de management a rețelei. Astfel, se pot minimiza costurile operaționale și se pot elimina perioadele de instabilitate apărute datorită erorilor inserate de către factorul uman.

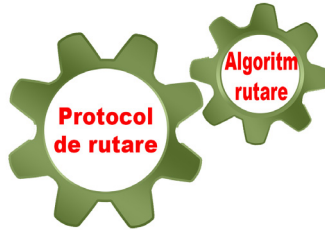
Mecanismele de tip SDN și în consecință tehnologia OpenFlow apare ca răspuns la tendințele de virtualizare a resurselor de calcul, dorința de îmbunătățire a mecanismelor de mobilitate și apariția soluțiilor de tip *IT-as-a-Service* (infrastructură IT ce este utilizată doar la nevoie), care exercită o presiune semnificativă, căreia rețeaua clasică nu îi poate face față. Astfel, este nevoie ca aceasta să se poată adapta în timp real la cerințele dinamice provenite din partea utilizatorilor, devenind astfel o platformă optimizată pentru livrarea eficientă a unei game variate de servicii.

## OPTIMIZAREA RUTARII UTILIZÂND COMUTAȚIA DE ALGORITMI

Nu putem vorbi de o rețea dinamică ce are capacitatea de a se adapta la necesitățile utilizatorilor și a serviciilor transportate, fără să abordăm diverse aspecte de eficientizare a *rutării*. În consecință, în acest capitol vom prezenta o soluție ce își propune să ofere posibilitatea de implementare a mecanismelor dinamice de *rutare*.

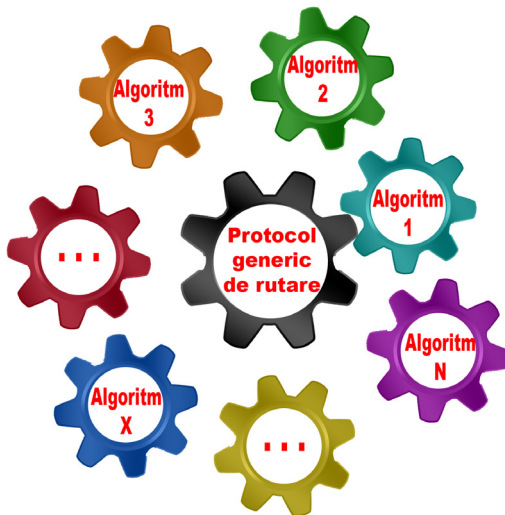
Paradigma după care a evoluat lumea rețelelor de calculatoare de-a lungul timpului presupunea elaborarea a câte unui protocol distinct pentru rezolvarea fiecărei probleme de comunicații. Se poate înțelege astfel motivul pentru care în ziua de azi exista o gamă largă de protocoale ce trebuie activate pe echipamentele din infrastructura rețelei în vederea implementării unui anumit set de politici. În ceea ce privește *rutarea*, există protocoale de *rutare* dedicate fiecărui scenariu ce poate fi întâlnit: protocoale de tip vector distanță, protocoale bazate pe starea legăturii, protocoale intradomeniu și protocoale ce se ocupă cu *rutarea* datelor între domenii autonome diferite. Fiecare dintre acestea folosec un anumit algoritm pentru luarea deciziilor de rutare (vezi *Figura 4*). Înțelegerea și utilizarea tuturor acestor soluții a devenit o sarcină dificilă pentru administratorii rețelelor, crescând astfel costurile operaționale

pentru astfel de infrastructuri.



*Figura 4. Relația dintre protocoalele de rutare și algoritmi în abordarea clasică*

În întâmpinarea acestei probleme, propunem unificarea tuturor protocoalelor de rutare într-unul singur ce poate fi utilizat în toate scenariile posibile. Acesta este capabil să își adapteze comportamentul în funcție de cerințele fiecărui flux de date în parte. Cu alte cuvinte, deciziile de *rutare* nu mai sunt luate de către un singur algoritm, indiferent de necesitățile curente. GRAS (Gearbox-like Routing Algorithms Selection) [8] este soluția propusă a fi răspunsul la problema enunțată anterior și presupune punerea la dispoziția protocolului de *rutare* a unui set de mai mulți algoritmi, din care va fi activat la un moment dat cel mai potrivit pentru scenariul în cauză (vezi *Figura 5*).



*Figura 5. Ilustrarea conceptului GRAS*



Algoritmii prevăzuți a fi utilizați într-un astfel de sistem adaptiv sunt următorii (setul poate fi extins în funcție de necesități cu efort minim, deoarece modificările ce trebuie aduse vor fi implementate doar pe elementul de control centralizat a rețelei cu capabilități OpenFlow):

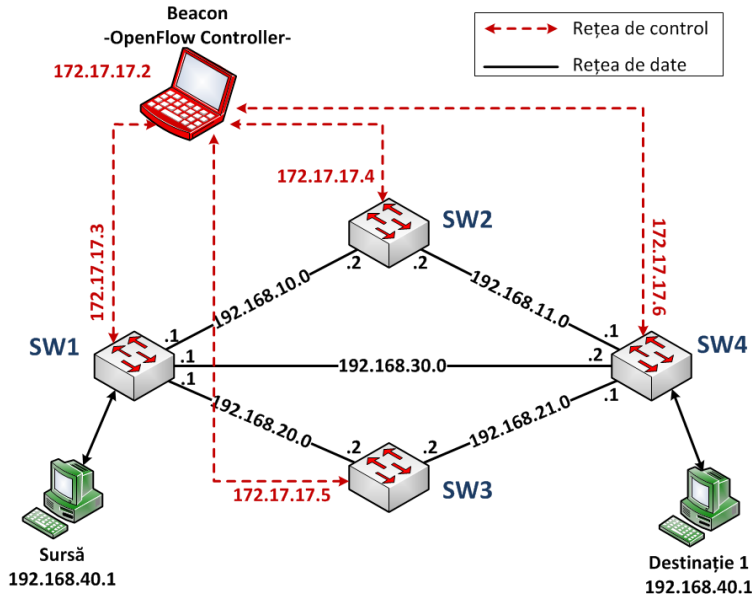
- a. Floyd-Warshall [9]: utilizat pentru găsirea căii optime între toate perechile de noduri sursă-destinație dintr-o topologie oarecare. Acest algoritm este recomandat a fi utilizat de către operatorii de infrastructură de telecomunicații pentru a oferi tuturor abonaților cea mai bună cale spre destinație.
- b. Dijkstra modificat [10]: este optimizat pentru găsirea căii optime dintre un nod sursă și mai multe noduri destinație într-o infrastructură dată. Acest model de comunicație este recomandat în scenariile în care se dorește găsirea căilor optime de la un furnizor de servicii (transmisii video la cerere, televiziune prin Internet, furnizor de servicii *cloud* etc.) către clienții acestuia.
- c. Ford-Fulkerson [11]: acest algoritm poate fi folosit cu succes în cazul în care ruta optimă dintre un nod sursă și un nod destinație nu are capacitatea de a suporta rata de transfer a datelor ce trebuie transmise între cele două noduri. În consecință, este necesară utilizarea acestui algoritm de *rutare* pe căi multiple pentru a găsi toate traseele disponibile între sursă și destinație.

În funcție de starea rețelei și a cerințelor de performanță provenite din partea utilizatorilor și a serviciilor transportate, se va alege algoritmul optim din lista de mai sus. Deoarece infrastructura de comunicații are proprietăți dinamice în timp, este nevoie ca măsurători specializate să se efectueze în timp real astfel încât deciziile luate să fie în concordanță cu starea actuală.

## REZULTATE EXPERIMENTALE

Pentru a ilustra fezabilitatea ideilor enunțate în capitolul anterior s-a utilizat o topologie de test formată din patru echipamente pe care rulează soluția Open vSwitch 1.3 (acesta este un *switch* virtual ce implementează tehnologia OpenFlow). Managementul în timp real al rețelei s-a făcut de pe un calculator adițional pe care rulează soluția Beacon (prezentat în capitolul 2) și utilizând o infrastructură de rețea dedicată schimbului de date de control. Traficul util trimis prin rețeaua de test a fost transmis între o stație cu

rol de sursă și una cu rol de destinație (vezi *Figura 6*).



*Figura 6. Topologia de test utilizată*

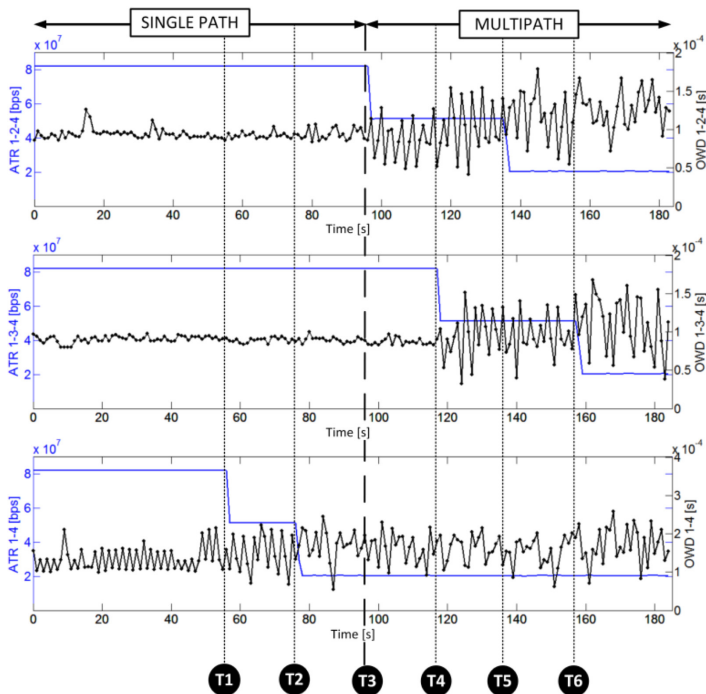
Nodurile SW1, SW2, SW3 și SW4 sunt configurate să efectueze comutația astfel: de fiecare dată când un pachet este recepționat, se încearcă determinarea fluxului din care face parte. Dacă o acțiune a fost definită vizavi de fluxul în cauză, aceasta va fi executată (pachetul va fi transmis pe portul de ieșire corespunzător, determinat cu ajutorul algoritmului de rutare activat pentru situația respectivă), în caz contrar, pachetul este încapsulat și trimis mai departe către entitatea de control (Beacon). După procesarea acestei informații se va trimite către nodul de comutație acțiunea ce trebuie aplicată pentru următoarele pachete primite, ce fac parte din acest nou flux.

Scenariul de test implementat presupune transmisia între nodurile Sursă și Destinație a șase fluxuri ( $F_1, F_2, \dots, F_6$ ) a câte 30 Mbps fiecare, către nodul Destinație. Acestea au fost pornite la un interval de 20 secunde unul de celălalt. Marcăm momentele de timp astfel:  $T_1, T_2=T_1+20s, \dots, T_6=T_5+20s$ . În acest scenariu s-a decis utilizarea algoritmului Dijkstra modificat pentru identificarea celei mai bune căi între sursă și destinație. Deoarece numărul de fluxuri crește în fiecare pas, se observă faptul că din pasul 3 elementul de control decide folosirea algoritmului Ford-Fulkerson pentru utilizarea în pa-

ralet a mai multor căi de transmisie a datelor. Rezultatul cu privire la alocarea fiecărui flux pe rutele existente, poate fi vizualizat în *Tabelul 1*.

Ruta	T1	T2	T3	T4	T5	T6
1-2-4	-	-	F3	F3	F3, F5	F3, F5
1-3-4	-	-	-	F4	F4	F4, F6
1-4	F1	F1, F2	F1, F2	F1, F2	F1, F2	F1, F2

*Tabel 1. Alocarea fluxurilor de date pe fiecare rută*



*Figura 7. Testul 1: ATR și OWD*

În *Figura 7* sunt ilustrate graficele ATR (Available Transfer Rate – rata de transfer disponibilă pe o anumită rută) și OWD (One-Way Delay – latența măsurată pe o rută) pentru cele trei căi existente în topologia de test. Se observă faptul că de la un moment la altul starea rețelei se modifică datorită

aparitiei de noi fluxuri de date trimise către destinație. Utilizând algoritmul Ford-Fulkerson se evită astfel utilizarea excesivă a unei singure legături din rețea (fenomen întâlnit la protocoalele de rutare clasice). Traficul este alocat dinamic pe toate căile disponibile din infrastructură, reușindu-se astfel întârzierea momentului de apariție a fenomenului de congestie.

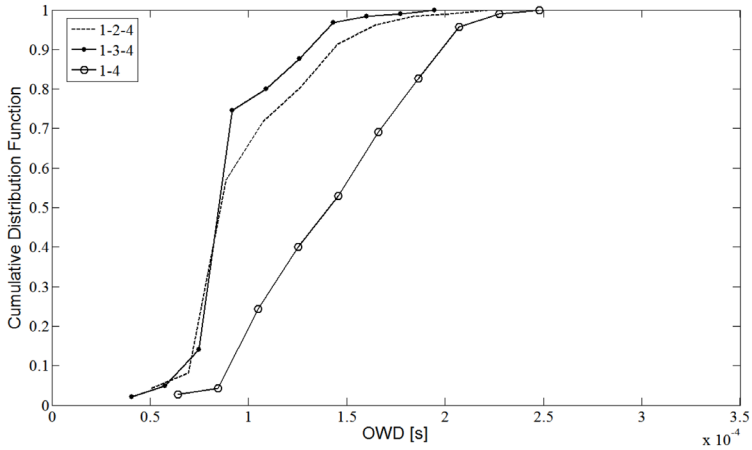


Figura 8. Funcția CDF pentru valorile OWD pe cele trei rute

Rezultatele ilustrate în *Figura 8* demonstrează faptul că implementându-se un comportament adaptiv al rețelei la situația actuală se pot menține parametri de calitate ai serviciilor (latența) la un nivel mulțumitor chiar și în cazul în care rețeaua începe să fie utilizată până aproape de capacitatea maximă. În consecință, acest aspect se va reflecta prin păstrarea unui grad ridicat de satisfacție a utilizatorilor infrastructurii de comunicații.

## CONCLUZII

Evoluția către o lume a rețelelor programabile, capabile să se adapteze în timp real la schimbările ce pot apărea, nu se poate face fără separarea planului de control de cel al comutației. O soluție viabilă în acest sens este OpenFlow, care reușește să abstractizeze funcționarea rețelei la un nivel la care o aplicație poate prelua controlul acesteia. În consecință, comportamentul rețelei va fi optimizat la cerințele existente indiferent de dinamicitatea acestora.

Scenariile ce pot beneficia de pe urma capabilității unei infrastructuri de a fi programabilă, sunt nenumărate. Printre cele de importanță ridicată se numără aplicațiile de optimizare a *rutării* în funcție de necesitățile reale ale utilizatorilor și ale serviciilor utilizate de către aceștia. În acest articol am demonstrat faptul că utilizând un protocol de rutare ce folosește adaptiv, în funcție de situația dată, un anumit algoritm de calcul a căilor optime, putem eficientiza modul de utilizare a resurselor, păstrând astfel un grad ridicat de satisfacție. Astfel de abordări vor putea fi testate și mai apoi implementate în rețelele de producție din viitor într-un timp relativ scurt, de îndată ce tehnologiile SDN vor fi adoptate la scară largă.

## REFERINȚE BIBLIOGRAFICE

- [1] N. Feamster, J. Rexford, E. Zegura, „The Road to SDN: An Intellectual History of Programmable Networks”, ACM SIGCOMM Computer Communications Review, Aprilie 2014.
- [2] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh,
- [3] S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hlzle, S. Stuart, and A. Vahdat, „B4: Experience with a globally deployed software defined WAN”, *ACM SIGCOMM*, Aug. 2013.
- [4] \*\*\* „It’s time to virtualize the network”, Nicira Whitepaper 2012, <http://nicira.com/en/network-virtualization-platform>
- [5] \*\*\* „Open Networking Foundation”, [www.opennetworking.org](http://www.opennetworking.org)
- [6] \*\*\* „Open Daylight”, <http://www.opendaylight.org/>
- [7] \*\*\* „Software-Defined Networking: The New Norm for Networks”, Open Networking Foundation ONF White Paper, April 13 2012
- [8] N. McKeown, et al., “OpenFlow: enabling innovation in campus networks”, *ACM SIGCOMM Computer Communication Review*, 38 (2), April 2008, 69-74
- [9] A.B. Rus and V. Dobrota, “Case Study of a Gearbox-Like Routing Algorithm Selection in Runtime”, *18th IEEE LANMAN 2011*, Chapel Hill, North Carolina, pp. 1-6, DOI: 10.1109/LANMAN.2011.6076938
- [10] A.G. Furculita, M.V. Ulinic, A.B. Rus, and V. Dobrota, “Implementation Is-

- sues for Modified Dijkstra's and Floyd- Warshall Algorithms in OpenFlow", *12th RoEduNet International Conference*, Constanta, Romania, 2013, pp.50-55, DOI:10.1109/RoEduNet.2013.6714208
11. [10] A.B. Rus, V. Dobrota, A. Vedinas, G. Boanea, and M. Barabas, "Modified Dijkstra's algorithm with cross-layer QoS", *ACTA TECHNICA NAPOCENSIS, Electronics and Telecommunications*, vol. 51, no.3, 2010, pp. 75-80.
  12. [11] P. Sevastian, A.B. Rus & V. Dobrota, "Simulation of the Ford-Fulkerson Algorithm Using OMNET++", *ACTA TECHNICA NAPOCENSIS, Electronics and Telecommunications*, ISSN 1221-6542, Vol.53, No.2, 2012, pp. 23-29

### Autori

**Dr. ing. Bogdan Rus** - Șef lucrări @ Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei, Universitatea Tehnică din Cluj-Napoca

**Dr. ing Virgil Dobrota** - Profesor @ Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei, Universitatea Tehnică din Cluj-Napoca