

# Verificarea Rețelelor Programabile



**Dragoș Dumitrescu**

Coordonator: Prof. Dr. Ing. Dragoș Niculescu

Facultatea de Automatică și Calculatoare  
Universitatea Politehnica București

*Teză de Doctorat*

București

Aprilie 2022



## Rezumat

**Cuvinte-cheie:** rețele, verificarea modelelor, verificare, verificarea rețelelor, P4, SDN, BGP, corectitudinea rețelelor, rețele in cloud.

Rețele de calculatoare sunt în centrul vieților noastre moderne; ele constituie fundațiile oricăror sisteme de calcul. Așadar, ele trebuie să fie robuste și fără probleme (bug-uri) pentru a asigura funcționarea corectă a aplicațiilor critice pentru siguranță și pentru bunul mers al afacerilor. Creșterea în popularitate a programabilității în planurile de date și de control ale rețelelor a deschis drumul a numeroase aplicații cu numeroși utilizatori, a infrastructurilor de tip cloud. Acestea cresc complexitatea rețelelor, administrarea și depanarea acestora devenind din ce în ce mai anevoioasă. Pentru a rezolva această problemă, a apărut un întreg nou domeniu de cercetare – verificarea rețelelor. Acesta a înregistrat succese semnificative în asigurarea corectitudinii rețelelor în ultimii zece ani.

În această dizertație, prezint cercetarea pe care am întreprins-o în domeniul verificării rețelelor. În acest sens, m-am concentrat pe dezvoltarea unor soluții practice de verificare a rețelelor. Practicalitatea este explorată din două perspective: ușurința în utilizare și performanța. În ce privește ușurința în exploatare, am explorat proprietăți implicite de corectitudine care trebuie să țină pentru toate rețelele considerate; astfel, administratorul beneficiază de *zero* efort de specificare. În ce privește performanța rețelelor, principala cerință este ca uneltele de verificare să scaleze până la dimensiunile rețelelor dintr-un cloud public. Acesta constă de obicei din mii/zeci de mii de dispozitive de rețea, servere și o mare diversitate de aplicații de rețea.

Contribuțiile acestei teze se poziționează conform cu nivelurile de programabilitate întâlnite în rețelele unui operator de cloud și arată cum rezolv diversele probleme impuse la fiecare nivel. Programarea la nivelul dispozitivelor din planul de date (e.g., dispozitive P4) se ocupă cu programarea dispozitivelor de comutarea pachete; rețelele definite în software (SDN – software defined networks; e.g. OpenFlow) au ca obiectiv orchestrarea mai multor dispozitive, în timp ce configurarea la nivelul planului de control (e.g., politici de rutare) se ocupă de diseminarea informațiilor de control către toate dispozitivele administrate de operator și mai departe în Internet. Cercetarea mea răspunde la întrebări la toate aceste nivele. În primul rând, contribuțiile mele la Vera (Stoenescu et. al., Debugging P4 programs with Vera in SIGCOMM'18), Vera2

(Dumitrescu et. al., Dataplane verification for P4, prezentare în workshop-ul NetPL'19) și bf4 (Dumitrescu et.al. bf4: towards bug-free P4 programs în SIGCOMM'20) arată cum poate fi realizată detecția practică și ușor de folosit de bug-uri în programele P4. În al doilea rând, *netdiff* (Dumitrescu et.al. Dataplane equivalence and its applications în NSDI'20) este un algoritm care utilizează echivalența planelor de date pentru a detecta bug-uri în controllere SDN. A treia contribuție constă în scalarea simulării în rețele BGP ale unor operatori de cloud.

**Rezumatul contribuțiilor.** Poziționarea lucrărilor mele în literatura verificării rețelelor este un indicator pentru principala întrebare de cercetare urmărită pe parcursul acestei teze: "găsirea unor proceduri de decizie scalabile în contextul unor politici complexe și programabile de rețea".

Cel mai important principiu al lucrărilor mele constă în ușurința în specificare sau chiar absența completă a acesteia (zero-specificare). Lucrările mele atât în verificarea dispozitivelor programabile, a controllerelor SDN utilizează noțiuni implicite de corectitudine și croiesc algoritmi de decizie pe baza specificităților acestora.

Contribuțiile mele la verificarea în planul de control se bazează pe unelte care sunt specific create pentru a scala. Așadar, ele respectă în mod implicit principiul *zero-specificare*. Orice verificare nouă necesită o implementare specifică ce oferă garanții de performanță sporite.

Redau întrebarea de cercetare a acestei teze: "Este posibilă verificarea rețelelor programabile cu foarte puțin (poate chiar deloc) efort de specificare din partea utilizatorilor rețelei?"

Această teză conține material științific publicat anterior de autorul acestei teze în conferințe de specialitate. Secțiunile [4.1](#) și [2.6.3](#) vor fi publicate în viitorul apropiat.

**Conținutul tezei.** Începând de la cel mai de jos nivel al stivei OSI, în Capitolul [2](#), descriu contribuțiile mele la verificarea dispozitivelor programabile în P4. Section [2.5](#) prezintă un nou algoritm de inferență a unor filtre la rulare care să asigure absența bug-urilor din programe în P4. Secțiunea [2.6](#) oferă un raport tehnic detaliat despre proiectarea și implementarea uneltelor de verificare a programelor P4.

În capitolul [3](#), detaliez echivalența planelor de date, o formă implicită de specificare a corectitudinii. Observația este că rețelele virtuale sunt adeseori privite din perspective distincte – intenția clientului vs. ce este de fapt implementat de operator. Aceste perspective sunt ținute sincronizat de un middle-ware complex și distribuit. Acesta este susceptibil la bug-uri și inconsecvențe. Scopul acestui capitol este de a arată corectitudinea prin verificarea echivalenței între cele două perspective. Folosind această metodă, am găsit bug-uri rare într-o instalare a Neutron – componenta de rețea a software-ului OpenStack.

Capitolul 4 prezintă contribuțiile mele la simularea planului de control în rețelele operatorilor de cloud. În secțiunea 4.1, descriu metodele de a scala simularea rețelelor BGP în scenariul unor centre de date de nivel de producție.

Capitolul 5 trage concluziile acestei dizertații și le pune în contextul întrebării de cercetare formulate mai sus. Tot aici sunt prezentate și perspectivele de continuare a cercetării raportate în aceasta lucrare.



# Cuprins

<b>1</b>	<b>Introducere</b>	<b>1</b>
1.1	Programabilitate în rețele cloud . . . . .	2
1.2	Găsirea bug-urilor în rețele . . . . .	7
1.3	Rezumatul contribuțiilor . . . . .	10
1.4	Literatura în domeniul verificării rețelelor . . . . .	11
<b>2</b>	<b>Verificarea programelor P4</b>	<b>15</b>
2.1	Introducere în limbajul P4 . . . . .	15
2.2	Bug-uri în programele P4 . . . . .	18
2.3	Depanarea programelor P4 cu Vera . . . . .	19
2.3.1	Traducerea programelor P4 în SEFL . . . . .	19
2.3.2	Evaluare . . . . .	23
2.4	Vera2 – Generarea condițiilor de verificare pentru programe P4 . . . . .	26
2.5	bf4 – înspre programe P4 fără bug-uri . . . . .	28
2.5.1	Introducere . . . . .	28
2.5.2	Motivație . . . . .	30
2.5.3	Soluții . . . . .	32
2.5.4	bf4 – vedere de ansamblu . . . . .	34
2.5.5	Evaluare . . . . .	48
2.5.6	Lucrări înrudite . . . . .	54
2.5.7	Concluzii . . . . .	54
2.6	Anatomia unei unelte de verificare P4 . . . . .	55
2.6.1	Introducere . . . . .	55
2.6.2	Ingredientele necesare verificării programelor P4 . . . . .	56
2.6.3	Tangibilitate cu primitive de procesare pachete . . . . .	72
2.6.4	Pași următori . . . . .	78

---

<b>3</b>	<b>Echivalența și aplicațiile ei în verificarea planului de date al rețelei</b>	<b>81</b>
3.1	Introducere . . . . .	81
3.2	Motivație . . . . .	83
3.3	Abordări în verificarea echivalenței . . . . .	85
3.3.1	Soluțiile existente sunt insuficiente . . . . .	86
3.4	Echivalența panelor de date cu netdiff . . . . .	87
3.4.1	Execuția simbolică a panelor de date . . . . .	89
3.4.2	Echivalența între mulțimi de căi . . . . .	90
3.4.3	Corectitudine si complexitate . . . . .	92
3.5	Corectitudinea netdiff . . . . .	93
3.6	Implementare . . . . .	95
3.6.1	Integrarea cu OpenStack Neutron . . . . .	96
3.7	Evaluare . . . . .	98
3.7.1	Bug-uri în Neutron . . . . .	98
3.7.2	Verificarea unei instalări Neutron de dimensiuni mari . . . . .	103
3.7.3	Echivalența programelor P4 . . . . .	104
3.8	Lucrări înrudite . . . . .	105
3.9	Concluzii . . . . .	106
<b>4</b>	<b>Scalarea verificării în rețele cu BGP</b>	<b>109</b>
4.1	Optimizarea simulării rețelelor în centre de date . . . . .	110
4.1.1	Introducere . . . . .	110
4.1.2	Exemplu . . . . .	112
4.1.3	Algoritm . . . . .	120
4.1.4	Evaluare . . . . .	132
4.1.5	Lucrări înrudite . . . . .	133
4.1.6	Concluzii . . . . .	134
<b>5</b>	<b>Concluzii</b>	<b>137</b>
5.1	Oportunități de cercetare ulterioară . . . . .	139