



**UNIVERSITATEA POLITEHNICA
DE BUCURESTI**



**Scoala Doctorala de Electronica, Telecomunicatii
și Tehnologia Informației**

Decizia nr. ____ de la ZZ-LL-AAAA

Ph.D. TEZA REZUMAT

Mustafa Khaleel Hamadani

**TEHNOLOGII CLOUD IN RETEA 5G INTRE
DISPOZITIVE DE COMUNICATII**

**TEHNOLOGII DE CLOUD COMPUTING ÎN REȚELE
5G PENTRU COMUNICARE D2D**

COMISIA DE TEZĂ

prof. dr. ing. Eugen Borcoci
Politehnica Univ. al Bucurestiului

Ph.D. supraveghetor

BUCURESTI 2022

Cuprins

Cuprins.....	i
1 Introducere.....	1
1.1 Structura doctoratului. teză	2
2 Context și concepte aferente	2
2.1 Prezentare generală a comunicațiilor D2D.....	2
2.2 Tehnologii de colaborare: Cloud, Edge și Fog.....	3
2.3 Rețeaua definită de software (SDN).....	4
2.4 Aspect arhitectural.....	5
3 Fog Computing și cooperarea rețelelor D2D	6
3.1 Model matematic pentru cooperarea Fog-D2D.....	6
3.2 Model de cerere de servicii pentru clustere D2D	7
3.3 Modele Fog Nodes Network și Cloud Servers.....	7
3.3.1 Model de noduri de ceață (model de așteptare M/M/m/K).....	8
3.4 Algoritmul K-Medoid pentru nodurile de ceațăPlasarea.....	10
Tehnicile de echilibrare a sarcinii pentru rețelele de calcul în ceață	12
3.5 Formularea problemei de echilibrare a sarcinii.....	14
3.6 Algoritmul genetic pentru echilibrarea sarcinii.....	15
4 Rezultatele simulării.....	16
4.1 Scenariile de simulare a plasării nodurilor de ceață.....	16
4.1.1 Scenariul sarcinilor de serviciu de dimensiuni mici și mari	17
4.1.2 Scenariul de simulare K-Medoids.....	21

4.1.3	Concluzie	24
4.2	Echilibrarea sarcinii SDN pentru rețelele de noduri de ceață	25
4.2.1	Primul scenariu (diferite dimensiuni ale sarcinilor de serviciu)	27
4.2.2	Al doilea scenariu (implementarea diferitelor VM).....	31
4.2.3	Al treilea scenariu (VM-uri scăzute și ridicate).....	32
4.2.4	Concluzie	34
5	Discuții și lucrări viitoare.....	35
	Lista rapoartelor științifice	36
	Rezumatul contribuțiilor originale.....	36
	Publicații	37
	Referință	38

Capitolul 1

1 Introducere

Dezvoltarea accelerată a rețelei 5G și implementarea aplicațiilor care necesită mult resurse (de exemplu, jocurile 3D) necesită resurse suficiente pentru procesarea acestor aplicații cu o latență mai mică. Pe de altă parte, dispozitivele finale au resurse de calcul limitate, care sunt insuficiente pentru aplicațiile cu consum mare de resurse. Răspunsul la unde putem găsi resurse de calcul disponibile, însoțite de comunicare cu latență scăzută, se află în jurul comunicației de la dispozitiv la dispozitiv și al tehnologiilor cloud..

Tehnologiile cloud sunt paradigma de calcul centralizat are resurse de calcul aproape infinite și furnizarea de resurse la cerere. Tehnologiile cloud (inclusiv Fog și Edge) pot depăși problema resurselor de calcul menționată mai sus.

Comunicarea Device-to-Device (D2D) este considerată unul dintre factorii cheie pentru diversele servicii 5G, inclusiv serviciile de siguranță publică, partajarea conținutului, extinderea acoperirii, comunicarea vehiculului[1]. D2D este privit ca o abordare promițătoare pentru a transfera traficul de date în apropiere, cu o latență scăzută de comunicare.

În această teză, încercăm să integrăm tehnologiile cloud cu comunicarea D2D, unde setul utilizatorilor D2D este văzut ca o rețea de clustere D2D. Aceste clustere sunt considerate surse de date care generează o sarcină de servicii consumatoare de resurse. Tehnologiile cloud oferă o resursă de calcul pentru aceste servicii intensive în resurse.

Integrarea tehnologiilor cloud la comunicarea D2D ar putea fi ilustrată în Figura 1.2. utilizatorii D2D au format grupurile pe baza conținutului similar (de exemplu, streaming video). Utilizatorii D2D au generat o sarcină de service care va fi descărcată către nodurile de ceață prin stațiile de bază. Nodurile de ceață au primit sarcinile de service și le-au programat pentru mașinile virtuale adecvate, în funcție de categoria de sarcini de serviciu (de exemplu, jocurile online).

Prin urmare, vor ridica trei întrebări (printre alte probleme):

1. *Cum putem determina în mod optim numărul de noduri de ceață pentru a îmbunătăți criteriile specifice, cum ar fi limitele de întârziere?*
2. *Cum să plasați nodurile de ceață la anumite infrastructuri wireless (Stația de bază) pentru a minimiza distanța dintre stațiile de bază și nodurile de ceață. Astfel, întârzierea de acces între clusterelor D2D și nodurile Fog este redusă.*

3. *Cum putem distribui în mod optim sarcina de lucru între nodurile de calcul pentru a obține o cantitate aproape egală din volumul de lucru.*

1.1 Structura doctoratului. teză

Structura acestei teze este ilustrată după cum urmează:

Capitolul 1 titlul ca (introducere) motivează teza și evidențiază principalele probleme pe care le abordează această teză și întrebările de cercetare care se ridică.

capitolul 2 oferă o imagine de ansamblu asupra tehnologiilor de comunicare de la dispozitiv la dispozitiv și cloud abordate în teză.

capitolul 3 studiază problema de plasare a nodurilor de ceață, inclusiv determinarea numărului de noduri de ceață și a acestora în locații potențiale (stația de bază) din rețele.

capitolul 4a investigat distribuția sarcinii de lucru între nodurile de ceață (adică mașinile virtuale); algoritmul genetic adoptat ca soluție principală pentru problema echilibrării sarcinii.

capitolul 5a furnizat experimentele de simulare pentru ambele capitole 3 și 4.

Capitolul 6 discută rezultatele culese din capitolele precedente, evidențiază limitările soluției adoptate și sugerează lucrări viitoare.

capitolul 2

2 Context și concepte aferente

Acest capitol a oferit o privire de ansamblu asupra conceptelor conexe ale tehnologiilor incluse în această teză, aceste tehnologii incluzând Comunicarea D2D, tehnologiile cloud și SDN. Acest capitol a introdus o arhitectură care a integrat comunicația D2D și calculul Fog sub supravegherea SDN. Arhitectura menționată mai jos introduce un număr de probleme care vor fi discutate în capitolele următoare

2.1 Prezentare generală a comunicațiilor D2D

Comunicarea D2D este caracterizată ca comunicare directă între UE-uri în proximitatea unuia altuia cu implicarea completă/parțială a infrastructurilor celulare (Base Station). Mai mult decât atât, comunicarea D2D există în spectrul celular licențiat, ceea ce face ca D2D să ofere garanții QoS și detectarea perfectă a rețelei[2].

Proiectul de parteneriat de generație a 3-a (3GPP) a definit două proceduri pentru susținerea D2D în rețeaua celulară (LTE), și anume, D2D Discovery și D2D Communication. În descoperirea D2D, UE-urile au detectat și monitorizat alte UE-uri din apropiere. Descoperirea dispozitivului este clasificată în continuare în Descoperire centralizată și Descoperire distribuită[2].

Pentru Discovery centralizat, o entitate centralizată, cum ar fi stația de bază, va asista dispozitivele mobile în faza de descoperire. Mai mult, descoperirea centralizată, cunoscută și sub denumirea de descoperire asistată de rețea, poate fi clasificată în continuare în funcție de gradul de implicare a stației de bază în:

❖ *Implicarea completă a BS:*

În acest mod, stația de bază a controlat toate procedurile de descoperire a dispozitivelor, pornind de la dispozitivele reținute până la descoperirea inițială a celorlalte în apropiere.[2].

❖ *Implicarea parțială a BS:*

În această abordare, dispozitivele inițiază o descoperire a dispozitivului fără a obține nicio implicare din partea stației de bază[2]. Rolul stației de bază pornit începe în culegerea de informații despre calitatea canalului[2].

Pe de altă parte, Distributed Discovery, cunoscut sub numele de Direct Discovery, dispozitivele descoperă complet altele în apropiere, fără implicarea stației de bază.[2].

Principalele avantaje ale comunicării D2D ar putea fi rezumate după cum urmează:

- ❖ Comunicarea D2D oferă o dată mare și o latență scăzută (adică, câștig de proximitate) datorită legăturii directe dintre UE-urile.
- ❖ Eficiența spectrului s-a îmbunătățit odată cu creșterea numărului de biți transmiși în unitatea de lățime de bandă. În plus, eficiența energetică este îmbunătățită prin comprimarea mai multor date cu energie scăzută pe bit.

2.2 Tehnologii de colaborare: Cloud, Edge și Fog

Cloud Computing (CC) a luat în considerare infrastructura robustă care oferă calcularea și stocarea datelor de la distanță[3]. Cu toate acestea, distanța lungă dintre serverele cloud de la distanță și utilizatorii finali are ca rezultat o latență mare de transmisie care nu satisface cerințele stricte ale aplicațiilor moderne. Cloud Computing este o soluție centralizată care implică noduri centrale puternice care oferă servicii cloud utilizatorilor finali. Aceste servicii ar putea fi clasificate în Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) și Software-as-a-Service (SaaS)[4].

Alte două abordări sunt propuse pentru a depăși problema latenței: Fog și Edge computing. Paradigmele Fog și Edge împărtășeau un concept similar și le puteau folosi

interschimbabil[5]. Cu toate acestea, există mici diferențe între cele două; conform Open Fog Consortium, Fog computing este o paradigmă ierarhică și oferă calcul, rețele, stocare, unde Edge Computing este limitat doar la servicii de calcul[6]. O altă diferență legată de plasarea nodurilor, plasarea nodurilor de margine în cloud să fie la granițele furnizorilor de servicii (de exemplu, rețelele celulare și transportul inteligent), unde nodurile de ceață ar putea fi situate oriunde în apropierea nodurilor finale.[7].

2.3 Rețeaua definită de software (SDN)

Arhitectura tipică SDN este structurată în trei straturi/planuri diferite[8]: Planul de date, planul de control și planul de aplicare, așa cum este ilustrat în Figura 2.10. planul de date, cunoscut și sub denumirea de strat de redirectionare, este compus din dispozitive tipice de rețea fără nicio capacitate de luare a deciziilor, iar planul de control constă dintr-un nod centralizat responsabil de orchestrarea stratului de infrastructură de bază.

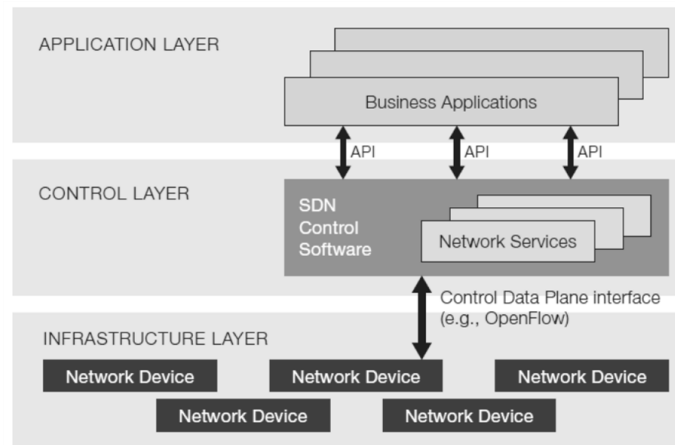


Figura2-1 Arhitectura SDN[8]

Planul de aplicare este desfășurat în partea de sus a planurilor anterioare; planul de aplicare realizează comportamentul rețelei și specifică politicile personalizate pentru managementul eficient al rețelelor. Ideea de bază a SDN este separarea dintre planul de control și planul de date. În SDN, dispozitivele de rețea, cum ar fi comutatoarele și routerele, au transmis pachete conform politicilor (regulilor) instalate în fiecare dispozitiv[8].

O cerință esențială a sistemului nostru propus este gestionarea unui număr mare de utilizatori de telefonie mobilă (adică, clustere D2D) și serviciile acestora pentru a satisface cerințele. Prin urmare, controlerul SDN din munca noastră este responsabil pentru gestionarea și controlul nodurilor de ceață, după cum urmează: *QoS*

- ❖ Controlerul SDN a colectat informații de rețea, inclusiv numărul de locuri de muncă în așteptare, timpul mediu de serviciu al serverelor de ceață, încărcarea conexiunilor de rețea.

- ❖ Controlerul SDN a construit deciziile optime de descărcare a sarcinilor de serviciu și a creat regulile de flux la stațiile de bază activate pentru SDN.
- ❖ Actualizarea serviciilor de găzduire, migrare și replicare a instanțelor de servicii; și reconfigurarea stației de bază în ceea ce privește regulile de flux[9].

După cum sa menționat mai devreme, planul aplicației a definit comportamentul rețelei și politicile personalizate pentru cazuri de utilizare specifice. În munca noastră, politicile de echilibrare a sarcinii implementate în planul aplicației pentru a valorifica gestionarea autonomă a resurselor nodurilor de ceață din rețea.

2.4 Aspect arhitectural

În această teză, introducem o arhitectură de sistem care se construiește pe baza lucrării[10], așa cum se arată în Figura 2.12. Arhitectura este formată din trei straturi; stratul de infrastructură a reprezentat diferitele infrastructuri wireless (de exemplu, stația de bază) și clustere D2D. Al doilea strat este rețeaua de noduri de ceață (FN) implementată la stațiile de bază și conectată la controlerul SDN (al treilea strat) conectate de asemenea la cloud pentru procesare ulterioară.

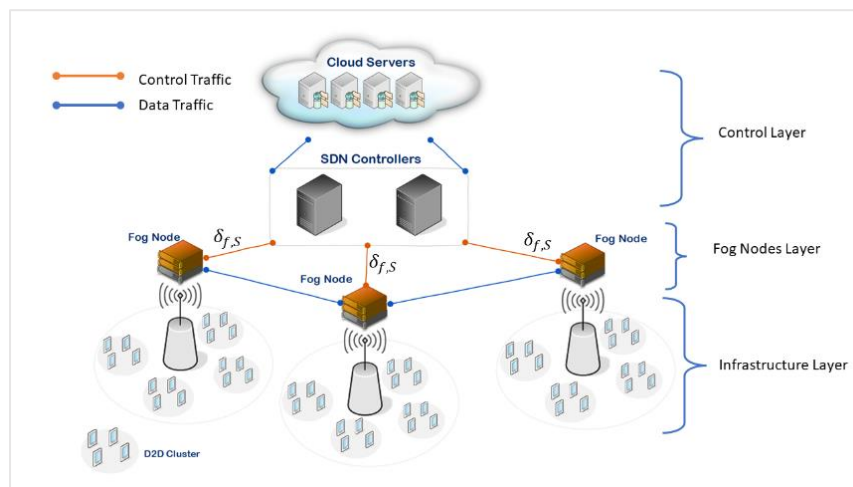


Figura2-2Aspectul arhitectural al sistemului

Al treilea strat cuprinde controlerul SDN și serverele de la distanță cloud, iar acest strat este considerat stratul de control. Clusterelor D2D sunt considerate sursa de date din sistem; generează o sarcină de serviciu (de exemplu, jocuri online), iar capul cluster este conectat la nodurile de ceață prin stațiile de bază. Aceste dispozitive sunt considerate dispozitive cu resurse limitate care generează o sarcină intensivă în resurse, iar aceste sarcini de serviciu se descarcă în nodurile de ceață pentru preprocesare. Nodurile Fog reprezintă serverele fizice care rulează diferite mașini virtuale (VM) care reprezintă diferite aplicații pentru cazuri de utilizator. În plus, acestea sunt eterogene în ceea ce privește puterea de calcul și capacitatea de stocare. *VMsVMs*

Controlerul SDN este responsabil pentru gestionarea și controlul nodurilor Fog prin stabilirea regulilor de flux care au definit politicile de echilibrare a sarcinii. Controlerul SDN monitorizează starea nodurilor de ceață prin colectarea de informații despre nodurile de ceață, inclusiv resursele de calcul disponibile, cantitatea de sarcini de serviciu în așteptare și altele. Pe baza acestor informații, controlerul a construit o vizualizare globală a rețelei și a definit algoritmul de echilibrare a sarcinii. Cu alte cuvinte, algoritmul de echilibrare a sarcinii va fi rulat central ca model de aplicație în arhitectura SDN.

În cele din urmă, serverul cloud la distanță (CRS) a reprezentat o sursă de resurse de calcul și stocare aproape nelimitate. CRS a procesat sarcinile de serviciu descărcate de la nodurile de ceață și a furnizat procesări suplimentare pentru controlerul SDN.

capitolul 3

3 Fog Computing și cooperarea rețelelor D2D

Acest capitol este compus din două părți pentru a studia problema implementării nodurilor de ceață (adică, numărul nodurilor de ceață și locația lor de implementare în rețea). În prima parte a acestui capitol sa concentrat pe calcularea numărului de noduri de ceață din rețea cu scopul de a minimiza timpul de răspuns al rețelei de noduri de ceață. Mai mult, a doua parte a determinat locațiile candidate (stația de bază) ale nodurilor de ceață din rețea, cu scopul de a minimiza distanța dintre stațiile de bază și nodurile de ceață. După determinarea numărului de noduri de ceață și a locației acestora în rețea. caracteristica nodurilor de ceață (numărul de mașini virtuale și atributele acestora), precum și problema echilibrării sarcinii vor fi discutate în capitolul 4.

3.1 Model matematic pentru cooperarea Fog-D2D

În această secțiune, prezentăm pe scurt un model matematic care descrie arhitectura introdusă în acea secțiune. Acest model matematic a fost construit pe baza lucrărilor propuse în[11].

Modelul matematic a fost construit pe baza teoriei cozii. După cum este prezentat în Figura 3.2, modelul de sistem constă din trei straturi: setul de clustere D2D, setul de straturi de noduri de ceață, controlere și serverele de la distanță cloud.

Considerăm o rețea celulară cu mai multe celule mici, iar fiecare celulă este acoperită de o stație de bază (BS) situată în centrul celulei mici. Considerăm un set al grupului de clustere D2D; ele generează cereri de servicii care vor fi trimise către nodurile Fog setate

prin stațiile de bază setate. $D = \{d_1, d_2, \dots, d_i\}$, $\{\lambda_1, \lambda_2, \dots, \lambda_i\}$, $F = \{f_1, f_2, \dots, f_m\}$, $\beta = \{\beta_1, \beta_2, \dots, \beta_n\}$

În plus, în cazul în care nodurile de ceață nu sunt capabile să gestioneze cererea de serviciu, aceasta este redirecționată către serverele de la distanță din cloud. Mai mult, fiecare stație de bază a gestionat și controlat mai multe clustere D2D notate ca α .

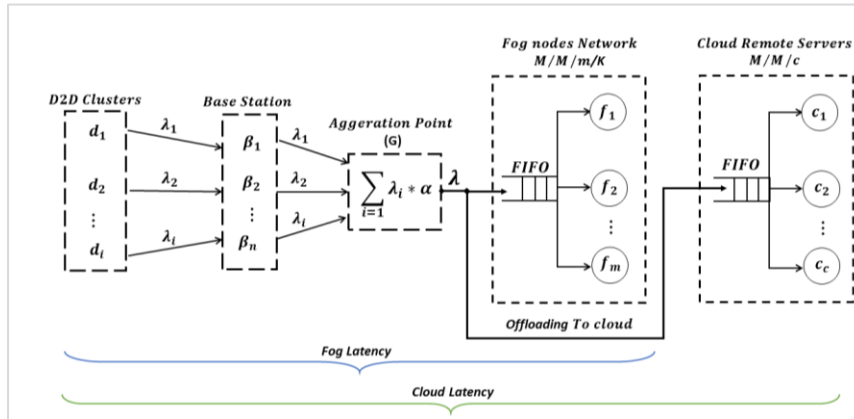


Figura3-1 Modelul sistemului (plasarea ceață)

Serverele cloud la distanță notate cu c , serverele cloud la distanță au putere de calcul și capacitate de stocare nelimitate. $C = \{c_1, c_2, \dots, c_c\}$

3.2 Model de cerere de servicii pentru clustere D2D

Clusterul D2D generează cerere de servicii. Mai mult, se presupune că aceste solicitări sunt distribuția Poisson (adică cererile independente unele de altele) $d_i \lambda_i$ [12]. În general, totalul cererilor de servicii (λ) agregate la ar putea fi definit ca: λG

$$Total\ Service\ Request\ (\lambda) = \sum_{i=1}^{\beta_n} \lambda_i \cdot \alpha \quad (3.1)$$

α : numărul de clustere D2D gestionate de fiecare stație de bază.

λ_i : Solicitarea serviciului de la un singur cluster D2D.

3.3 Modele Fog Nodes Network și Cloud Servers

În această secțiune, nodurile Fog și serverul Cloud sunt modelate ca un model de așteptare cu mai multe servere în care nodurile Fog sunt descrise ca, unde $M/M/m/K$ K a reprezentat capacitatea maximă a nodurilor de ceață. Serverele cloud sunt descrise ca $M/M/c$ cu capacitate nelimitată de a primi sarcini de servicii de la utilizatorii D2D. Scopul principal al acestei secțiuni este de a determina timpul de răspuns (adică timpul de calcul) pentru ambele modele de așteptare. Prin urmare, răspunsul total al sistemului este calculat.

3.3.1 Model de noduri de ceață (model de așteptare M/M/m/K)

Rețeaua de noduri Fog modelată ca model de așteptare cu mai multe servere. Unde a fost reprezentat numărul de servere fizice paralele identice care oferă un serviciu de ceață clusterelor D2D, unde a fost reprezentată capacitatea maximă a nodurilor de ceață de a primi sarcini de serviciu. Să presupunem că și sunt cererile de servicii de sosire și, respectiv, tariful de servicii. Prin urmare, dacă sosește o sarcină de serviciu pentru a găsi toate serverele ocupate, atunci sarcina este abandonată. Prin urmare, rata serviciului la acel moment este de $M/M/m/KmF = \{1,2, \dots m\}K\lambda\mu_F m\mu_F$

Pentru a calcula timpul de răspuns (adică timpul de calcul) al nodurilor de ceață, unele entități trebuie să determine, și anume (probabilitatea blocării) și numărul de clienți (sarcini de serviciu) din sistem. P_b

Probabilitatea de blocare () se referă și ca formula Erlang B sau formula de pierdere a lui Erlang și este adesea scrisă ca $P_b B(\lambda/\mu_F, m)$ [13].

$$P_b = \frac{(m\rho)^m / m!}{\sum_{k=0}^m \frac{(m\rho)^k}{k!}} \quad (3.2)$$

. Numărul de sarcini de servicii este notat ca și egal cu: (N_q)

$$N_q = \sum_{k=0}^m k P_k$$

$$N_q = (m\rho) \sum_{k=0}^m p_0 \frac{(m\rho)^k}{k!} = m\rho(1 - P_b) \quad (3.3)$$

In cele din urma,

Timpul de răspuns (dinul de calcul) este notat cu () poate fi calculat ca: T_F

$$T_F = \frac{m\rho(1 - P_b)}{\lambda} + \frac{1}{\mu} \quad (3.4)$$

După cum se vede din (3.10), ca număr de noduri de ceață() crește, celmtimp de raspuns (T_F) în scădere a rețelei; cu alte cuvinte, implementarea mai multor noduri de ceață în rețea va reduce timpul de procesare pentru serviciile furnizate utilizatorilor D2D. pentru a satisface cerințele QoS pentru utilizatorii D2D, am introdus un prag de timp QoS () de la locul de muncă T_{QoS} [14]. Mai mult, am restricționat timpul de răspuns () să fie egal sau mai mic decât (). $T_F T_{QoS}$

Deci, obiectivul nostru este de a determina numărul optim de noduri de ceață din rețea pentru a satisface cerința și s-ar putea defini astfel: QoS

$$\text{minimize } (T_F) \tag{3.5}$$

Sub rezerva:

$$T_F \leq T_{QoS} \tag{a} \tag{3.6}$$

$$m \in \mathbb{Z}^+ \tag{b}$$

3.12 (a) definită ca o constrângere ca timpul de răspuns să fie egal sau mai mic decât timpul) introdus din muncă $QoS(T_{QoS}$ [15]. definit ca o constrângere ca numărul de noduri de ceață să fie un număr întreg pozitiv (reprezentat numărul întreg pozitiv). pentru a rezolva ecuația (3.11), am propus un algoritm (vezi Algoritmul 3.1) pentru a obține numărul necesar de noduri de ceață. 3.12 (b) \mathbb{Z}^+

Algoritm 3.1 Calculul numărului de noduri de ceață

Intrare: cerere-serviciu, număr de BS, număr de cluster D2D, rata de serviciu, număr inițial de nod de ceață, noduri de ceață maxime și $\lambda_i \beta \alpha \mu_{FC} F_n F_n \text{Max } T_{QoS}$

Ieșire: , număr optim de noduri de ceață , $BestTps F_n Best\rho$.

$F_n = \text{inital value}$

$T_R^F = \text{inf}$

1. În timp ce $F_n < F_n \text{Max}$
2. calculați rata totală de sosire ecuația lambda (1) λ
3. calculează factorul de utilizare ρ

daca mai continua $\rho < 1$

altfel

$F_n = \text{treceți la pasul } 3F_n + 1,$

4. calculați formula erlang C
5. calculați ecuația (9) $Tps,$

dacă decât $Tps < T_{QoS}$

$BestTps = Tps$

$FnBest = Fn$

altfel

$Fn = Fn + 1$

treceți la pasul 1

6. sfârșitul în timp ce
7. **Întoarce-te și $.BestTpsFnBest$**

3.4 Algoritmul K-Medoid pentru nodurile de ceață Plasarea

După ce a fost determinat numărul de noduri de ceață, nodurile de ceață trebuie să fie plasate/desfășurate în anumite locații, astfel încât distanța dintre stația de bază și nodurile de ceață să fie scurtată. Prin urmare, întârzierea de acces între clusterul D2D și nodurile Fog va fi minimizată.

Arhitectura sistemului (vezi Figura 3-3) constă dintr-un set din grupul de clustere D2D; ele generează cereri de servicii care vor fi descărcate către nodurile Fog setate prin stații de bază setate care au fost implementate într-o anumită locație din rețea. Setul de noduri de ceață instalat la stația de bază este setat pentru a furniza servicii de ceață utilizatorilor de margine (clustere D2D). $D = \{d_1, d_2, \dots, d_i\}$ $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_i\}$ $F = \{f_1, f_2, \dots, f_m\}$ $\beta = \{\beta_1, \beta_2, \dots, \beta_n\}$ $F\beta$

Arhitectura sistemului ar putea fi definită ca un grafic nedirecționat, unde și este setul de stații de bază și este setul de locație potențială a nodurilor de ceață setate așa cum se arată în Figura 3.2. $G = (V, E)$ $V = \beta \cup K$ $\beta K = \{K_1, K_2, \dots, K_m\}$ F

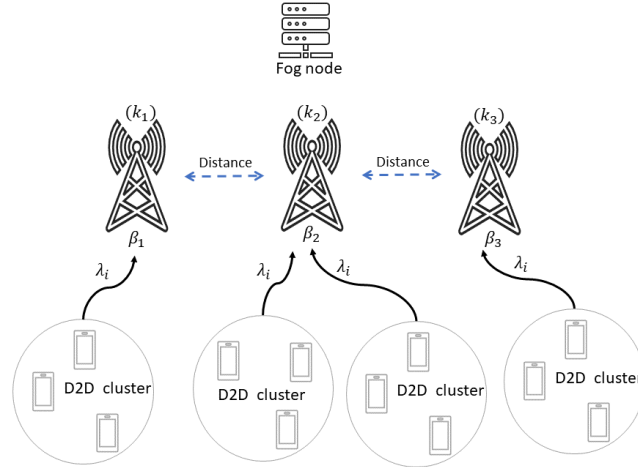


Figura3-2 Modelul Fog Nodes Placement

Acolo unde este reprezentat setul de conexiuni dintre o stație de bază și un nod de ceață, conexiunile dintre stațiile de bază și legăturile dintre dispozitivele mobile și stațiile de bază sunt în afara domeniului nostru de aplicare. E

Distanța euclidiană pătrată va fi considerată o măsurătoare între stația de bază și rețeaua nodurilor de ceață. Astfel, obiectivul nostru este să obținem o întârziere minimă de acces între stația de bază și nodurile Fog:

$$\text{Min } D(C_{\beta}, C_f) \quad (3.7)$$

Sub rezerva următoarelor constrângeri:

1. $x_{\beta f}$: decizia binară indică dacă o stație de bază este alocată nodului de ceață $\beta_n f_m$

$$x_{\beta f} = \begin{cases} 1, & \beta_n \text{ is assign to } f_m \\ 0, & \text{otherwise} \end{cases}$$

2. Două noduri Fog nu pot fi instalate la aceeași stație de bază.
3. Doar o stație de bază se conectează la nodul One Fog.

Tehnica de grupare K-medoids va fi adaptată pentru plasarea nodurilor de ceață. În K-medoids, rețeaua de stații de bază este împărțită în K zonă (sau regiune), unde K reprezintă numărul de noduri de ceață calculat mai devreme (vezi secțiunea 3.3.2 și algoritmul 3.1).

Considerăm o distanță euclidiană ca o măsurătoare pentru a defini granița clusterului și definită după cum urmează:

$$D_{Euclidean}(C_{\beta}, C_f) = \sqrt{(C_{\beta_1} - C_{f_1})^2 + (C_{\beta_2} - C_{f_2})^2} \quad (3.8)$$

Unde și sunt coordonatele de locație ale stației de bază și, respectiv, nodurilor de ceață. C_β, C_f

În K-medoid, un medoid poate fi definit ca punctul din cluster, cu o sumă minimă de distanțe până la toate celelalte puncte (vezi Figura 3.3)[16]. Distanța minimă dintre medoid (în cazul nostru va fi nodurile de ceață) și punctul clusterelor (stația de bază) se calculează după cum urmează:

$$D_{medoid}(C_\beta, C_f) = \sum_{C_{\beta n}} \sum_{C_{f m}} |C_\beta - C_f| \quad (3.9)$$

Algoritmul de grupare K-medoid ar putea fi descris ca următorii pași:

1. *Pasul 1 (inițializare)*
 - Alegeți puncte ca medoide inițiale. K
 - Calculați măsurarea distanței (în cazul nostru Distanța euclidiană pătrat).
 - S-au definit clusteri inițiale prin atribuirea fiecăruia celor mai apropiate medoide.
2. *Pasul 2 (selectarea Medoids)*

Selectați medoidele inițiale ale fiecărui cluster, care este punctul care are distanța minimă până la toate punctele din același cluster.
3. *Pasul 3 (Actualizare Medoids)*

Schimbați medoidele anterioare (de la pasul 3) cu toate celelalte puncte din grupul curent. Calculați suma distanței dintre puncte; noile Medoide vor fi punctele care au o distanță minimă față de toate celelalte puncte din clusterul actual. Figura 3.4 oferă o diagramă a pasului K-Medoids menționat mai sus.

capitolul 4

Tehnicile de echilibrare a sarcinii pentru rețelele de calcul în ceață

Acest capitol studiază problema de echilibrare a sarcinii care apare din cauza distribuției inegale a sarcinii de lucru între nodurile de ceață. se va realiza studiul caracteristicilor nodurilor de ceață (mașini virtuale și atributele acestora). problema de echilibrare a sarcinii poate fi definită ca o problemă de optimizare cu scopul de a minimiza execuția sarcinilor de serviciu primite de la clusterelor D2D. obiectivul principal al acestui capitol este distribuirea volumului de lucru aproape egal între nodurile de ceață. Fog computing se implementează ca noduri de ceață (adică, servere de ceață) în anumite regiuni din rețea; fiecare nod de ceață oferă resurse de calcul și capacități de stocare utilizatorilor de vârf.

Operația de distribuire a solicitărilor de servicii ale utilizatorilor între toate nodurile de ceață, astfel încât niciun nod de ceață să nu fie supraîncărcat sau subîncărcat (adică, inactiv) se numește Load Balancing (LB). Load Balancing este una dintre problemele de bază în tehnologiile cloud computing; Scopul echilibrării sarcinii asigură că toate nodurile de procesare (de exemplu, nodurile Fog) au primit aproximativ o cantitate egală de sarcină de lucru[17].

Modelul de sistem ar putea fi considerat ca un grafic nedirecționat (vezi Figura 4-2), unde este nod-set. Unde reprezentat setul de noduri de ceață, setul de controlere SDN și setul de servere cloud la distanță și . a prezentat setul de margini și a notat ca legătură între două noduri și noduri. $V = \{F, S, C\}$ $FF = \{f_1, f_2, \dots, f_m\}$ $S = \{s_1, s_2, \dots, s_j\}$ $CC = \{c_1, c_2, \dots, c_c\}$ $E = \{e_{a,b}\} e_{a,b} ab$

Mai mult, denotat ca latența de comunicare între și, respectiv, noduri. Mai mult, nodul de ceață (de exemplu,) conține un număr de mașini virtuale, prin urmare, numărul total dintre toate nodurile de ceață notate ca , , fiecare () are o capacitate limitată de a calcula resurse, cum ar fi CPU, memoria și capacitatea de stocare, și fiecare este descris cu următoarele atribute $\delta_{a,b} a b f_1 VMsN = \{VM_1 VM_2, \dots, VM_w\} VM_v VM_v \in NVM_v$ [18]:

$$VM_v = \{VM_v^{Id}, VM_v^{MIPS}\}$$

Unde este indicat identificatorul și a fost prezentată capacitatea de procesare a . În plus, capacitatea de procesare a notat ca $VM_v^{Id} VM_v VM_v^{MIPS} VM_v VM_v P_v$.

Clusterelor D2D generează o sarcină de serviciu computațională denumită {}, fiecare sarcină de servicii (sarcini de serviciu) poate fi descrisă după cum urmează $\{D_1, D_2, \dots, D_i\} \lambda_1, \lambda_2, \dots, \lambda_i \lambda_k \lambda_k \in [18]$:

$$\lambda_k = \{\lambda_k^{Id}, \lambda_k^{Length}\}$$

Acolo unde este indicat identificatorul sarcinii de serviciu, a prezentat durata sarcinii declarată ca instrucțiuni pe secundă. Mai mult, timpul estimat pentru finalizarea sarcinilor poate fi calculat după cum urmează $\lambda_k^{Id} \lambda_k^{Length} \lambda_k \lambda_k VM_v$ [19]:

$$ETC_{k,v} = \frac{\lambda_k^{Length}}{P_v} \quad (0.1)$$

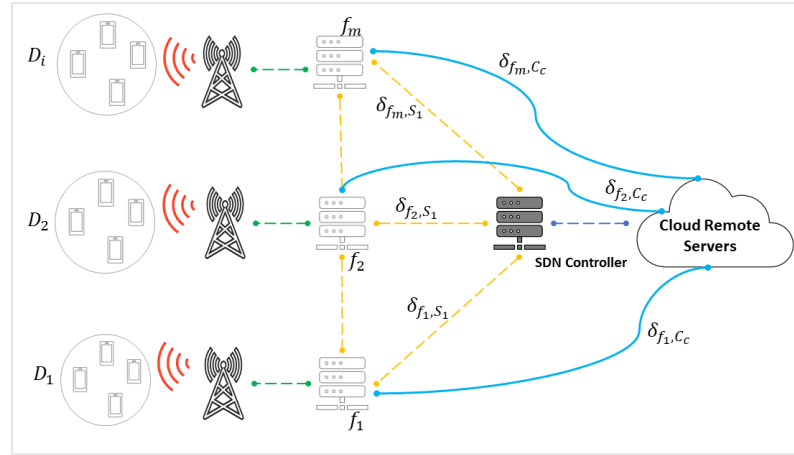


Figura0-1 Model de sistem (cu trei noduri de ceață și un controler SDN)

Am introdus o matrice (Expected Time to Compute) pentru a descrie modelul de sarcini pentru tehnologiile cloud în mediul eterogen *ETC* [20]. Matricea a ilustrat timpul așteptat pentru a finaliza/calcula o anumită sarcină de serviciu, așa cum se arată în (4.2). *ETCVM*

$$ETC = \begin{bmatrix} ETC_{1,1} & ETC_{1,2} & \cdots & ETC_{1,w} \\ ETC_{2,1} & ETC_{2,2} & \cdots & ETC_{2,w} \\ ETC_{3,1} & ETC_{3,2} & \cdots & ETC_{3,w} \\ \vdots & \vdots & \cdots & \vdots \\ ETC_{k,1} & ETC_{k,2} & \cdots & ETC_{i,w} \end{bmatrix} \quad (0.2)$$

În matrice, rândul matricei indică sarcina de serviciu () și coloana prezintă De exemplu, a exprimat timpul necesar pentru a finaliza o sarcină pe , a exprimat timpul necesar pentru a finaliza o sarcină și așa mai departe. $ETC_{\lambda_k} VM_v$. $ETC_{1,1} \lambda_1 VM_1$ $ETC_{3,2} \lambda_3 VM_2$

Prin urmare, scopul acestei lucrări este de a găsi o mapare optimă a sarcinii de serviciu astfel încât sarcina să fie aproximativ plasată pe toate nodurile și unul sau mai multe obiective îndeplinite. Astfel, se realizează echilibrarea sarcinii. *VM*

3.5 Formularea problemei de echilibrare a sarcinii

Problema de echilibrare a încărcăturii poate fi descrisă ca atribuirea sarcinii de serviciu mașinii virtuale, iar problema ar putea fi formulată ca o problemă de optimizare care a îndeplinit unul sau mai multe obiective. Cel mai frecvent obiectiv studiat în literatură este makespan (), care este definit ca „timpul maxim de execuție între toate VM-

urile” $k_v MS$ [21]. Pentru a calcula makespan-ul, în primul rând; timpul de execuție () al tuturor a fost calculat: $ETVM$

$$ET_{kv} = \sum_{k=1}^i \sum_{v=1}^w x_{k,v} \cdot ETC_{k,v} \quad (0.3)$$

Unde (Sarcini de serviciu și) și (, și și). $k \in 1 < k < i$ $v \in w1 < v < w$

$x_{k,v}$ este o variabilă de decizie inculpată dacă o sarcină de serviciu este alocată unui anumit sau nu. , dacă sarcina de service este alocată și altfel $VMx_{k,v} = 1$ $\lambda_k VM_v x_{k,v} = 0$.

Apoi, poate fi calculat ca: MS

$$MS = maximum (ET_{kv}) \quad (0.4)$$

În cele din urmă, problema de echilibrare a sarcinii pentru calculul de ceață poate fi formulată astfel:

$$minimize (MS) \quad (0.5)$$

Scopul problemei de echilibrare a sarcinii este de a găsi o mapare optimă care să minimizeze sarcina maximă. Complexitatea mapării sarcinii la egală cu (), prin urmare; Sarcina de echilibrare a sarcinii în calculul de ceață este considerată o problemă de împachetare cunoscută sub numele de grupuri NP-hard și NP-complete $i w VM w^i$ [22].

3.6 Algoritmul genetic pentru echilibrarea sarcinii

Algoritmii de aproximare (de exemplu, algoritmii genetici) sunt considerați o abordare alternativă pentru tehnicile de căutare exhaustive în soluții spațiale exponențial mari. Algoritmii genetici (GA) sunt algoritmi metaeuristici inspirați de teoria evoluției naturale a lui Charles Darwin.

În general, algoritmul genetic ar putea fi rezumat după cum urmează:

❖ *Populatia:*

un set de indivizi reprezintă o populație, fiecare individ din populație este considerat un cromozom.

❖ *Selectia cromozomilor:*

selectați indivizii (adică, părinții) din populația actuală pentru o soluție intermediară.

❖ *Crossover-ul:*

Operația de încrucișare poate fi rezumată prin selectarea a doi indivizi (considerați ca părinți) pentru a produce un nou individ numit copiii care s-au prezentat pentru următoarea generație.

❖ *Mutația:*

Operația de mutație modifică genele cromozomilor pentru a forma o nouă generație. Astfel, algoritmul este capabil să producă o soluție mai bună decât a fost calculată anterior.

❖ *Funcția de fitness:*

Funcția de fitness a măsurat corectitudinea indivizilor din populație. Prin urmare, indivizii supraviețuiesc sau încetează în funcție de condiția sau de valoarea funcției.

capitolul 5

4 Rezultatele simulării

Acest capitol oferă rezultatele implementării și simulării pentru problemele menționate în capitolele anterioare. Prin urmare, acest capitol este împărțit în două subcapitole.

Prima parte discută rezultatele simulării pentru capitolul 3 (determinați numărul de noduri de ceață și locațiile de implementare), MATLAB considerat ca instrument de simulare în această parte.

Partea a doua a discutat rezultatele simulării pentru capitolul 4 (echilibrarea sarcinii pentru rețeaua Fog Nodes), CloudSim considerat ca instrument de simulare în această parte.

4.1 Scenariile de simulare a plasării nodurilor de ceață

În această secțiune, sunt efectuate două scenarii de simulare pentru a studia și analiza integrarea calculului de ceață în rețelele D2D. Experimentele de simulare au fost implementate cu MATLAB.

4.1.1 Scenariul sarcinilor de serviciu de dimensiuni mici și mari

În acest scenariu, luăm în considerare două tipuri diferite de sarcini de serviciu (dimensiuni mici și mari ale sarcinilor) care sunt generate de clusterelor D2D. Sarcinile de service mici () au o dimensiune de aproximativ 2500 de instrucțiuni pe secundă, iar dimensiunea sarcinilor de service mari de aproximativ 5000 de instrucțiuni pe secundă. Am considerat că un număr total de clusterelor D2D în rețea este de aproximativ 1000 cluster și fiecare stație de bază gestionează și controlează un grup de clusterelor variat de la 10 la 40 de clusterelor (adică, 10, 20, 30 și 40). Prin urmare, numărul de clusterelor D2D a variat de la 250, 500, 750, 1000 de clusterelor, așa cum se arată în Tabelul 5.1. λ_i

Masa 4.1 Solicitățile totale de servicii

Numărul de cluster D2D	λ_i (În al doilea)	Solicitarea totală de servicii (λ)
250	0,25 req/s	62req/s
	0,5 req/s	125req/s
500	0,25 req/s	125req/s
	0,5 req/s	250req/s
750	0,25 req/s	187 req/s
	0,5 req/s	375req/s
1000	0,25 req/s	250 req/s
	0,5 req/s	500req/s

Rata de serviciu a nodului de ceață, aproximativ 50 de solicitări pe milisecundă, are și o dimensiune limitată a cozii de așteptare (vezi capitolul 3 secțiunea 3.3.2). De asemenea, am stabilit numărul maxim de noduri de ceață cu aproximativ 16 noduri, Tabelul 5.2 listele parametrilor utilizați în acest scenariu de simulare.

În acest studiu, mai multe valori de performanță sunt luate în considerare, după cum urmează:

1. *Timpul de răspuns.*

Această măsurătoare măsoară timpul necesar procesării/calculării cererii individuale. Poate fi determinată ca suma timpului de serviciu și a timpului de așteptare (vezi ecuația 10).

2. *Factorul de utilizare (ρ):* Această valoare reprezintă raportul dintre numărul de solicitări de sarcini/masaje și rata de servicii. Mai mult, factorul de utilizare (ρ) considerat ca o măsură a stabilității sistemului, iar valoarea sa ar trebui să fie mai mică de 1.

3. *Sarcina de lucru a sistemului*: măsurăm sarcina de lucru a rețelei nodului de ceață sub un număr diferit de cluster D2D pe o anumită perioadă de timp. Această măsurătoare calculează prin multiplicarea factorului de utilizare 100.

Efectul numărului de noduri de ceață asupra stabilității sistemului este ilustrat în figurile 5.2 și 5.3. În ambele figuri, sistemul a ajuns la stabilitate pe măsură ce numărul nodurilor de ceață crește. Înțelegem prin stabilitatea sistemului care trebuie să fie mai mică de $1.\rho$

Masa4.2 Parametri de simulare (primul scenariu)

Parametrii de simulare	Descriere
Cluster D2D (α)	10, 20, 30 și 40 cluster
Stația de bază (BS)	25 BS
Cereri de servicii λ_i	0,25 și 0,5 req/s
Tarif serviciului μ	50 req/s
Nodul maxim de ceață $FnMax$	16 noduri
T_{QoS}	0,4 s
Capacitate maximă a nodului de ceață (k)	254

Cu toate acestea, cererea totală de servicii afectează determinat numărul de noduri de ceață pentru a ajunge la stabilitate. λ

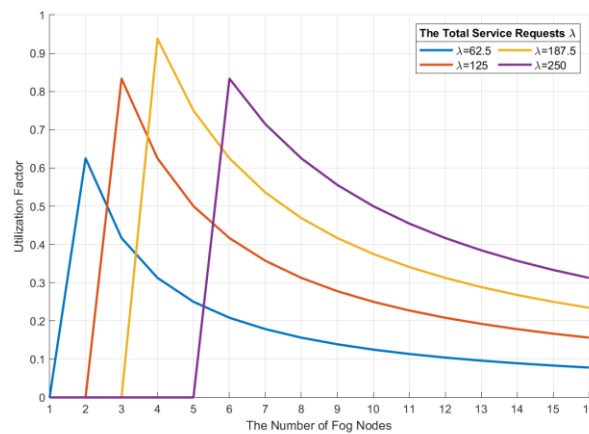


Figura4-1 Stabilitatea sistemului (sarcini de serviciu mici)

De exemplu, când cererea totală de servicii este egală cu 62 (sarcini de service de dimensiuni mici); sistemul are nevoie de aproximativ 2 noduri de ceață pentru a atinge stabilitatea (vezi Figura 5.2), iar pe de altă parte, când cererea totală de servicii este egală

cu 125, avem nevoie de aproximativ 3 pentru a ajunge la stabilitatea sistemului, așa cum se arată în Figura 5.2.

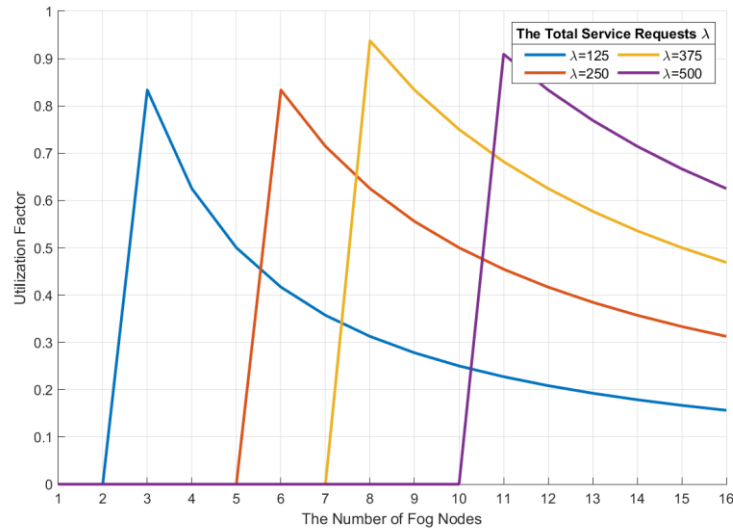


Figura4-2Stabilitatea sistemului (sarcini de service mari)

În mod similar, atunci când cererea totală de servicii (sarcini de serviciu de dimensiuni mari) este egală cu 250 (figura 5.3), numărul necesar de noduri de ceață este de aproximativ 6. În plus, numărul necesar de noduri de ceață pentru a ajunge la un sistem stabil este egal cu 11 în cazul în care cererea totală de servicii este egală cu 500.

După cum este ilustrat în Figura 5.4, numărul de noduri de ceață a crescut, timpul de calcul a scăzut; de exemplu, la cerere, avem nevoie de aproximativ 9 noduri de ceață pentru a atinge pragul () și motivul din spatele acestui număr specificat că sistemul între 1 și 8 este instabil () așa cum este ilustrat în figura 26. $\lambda = 250 T_{QoS} \rho = 0$

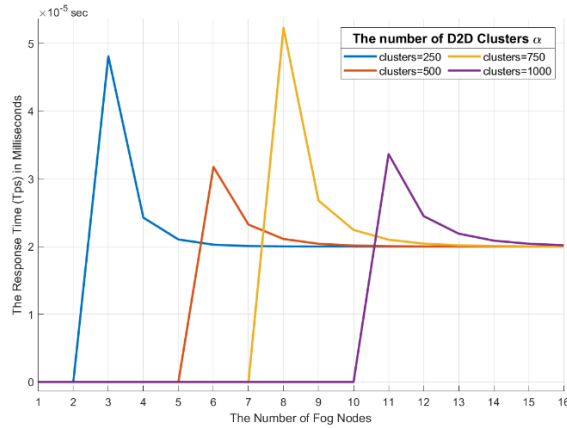


Figura4-3 Timpul de răspuns al rețelelor de noduri Fog (sarcini de serviciu de dimensiuni mari)

Figura 5.6 și 5.7 arată încărcătura de lucru în rețea sub diferite densități de clustere D2D cu diferite dimensiuni ale sarcinilor de serviciu.

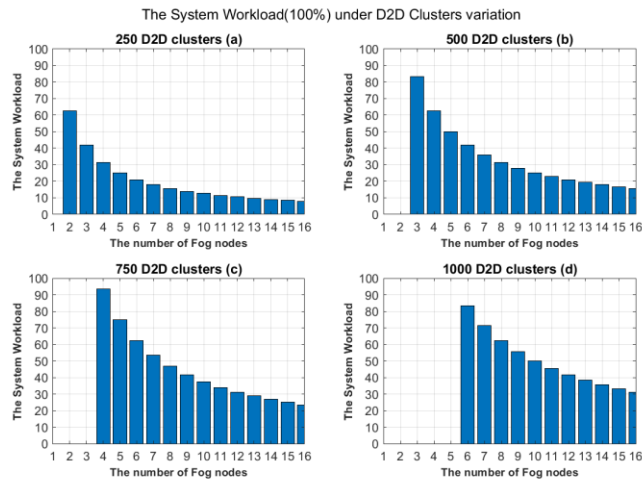


Figura4-4 Sarcina de lucru a sistemului cu diferite clustere D2D Densitate (sarcini de service de dimensiuni mici)

După cum observăm în Figura 5.2 (a) sarcina de lucru scade pe măsură ce numărul de noduri de ceață scade; ajungând la 45% și cu 3 noduri și continuă să ajungă la mai puțin de 40% cu 4 noduri.

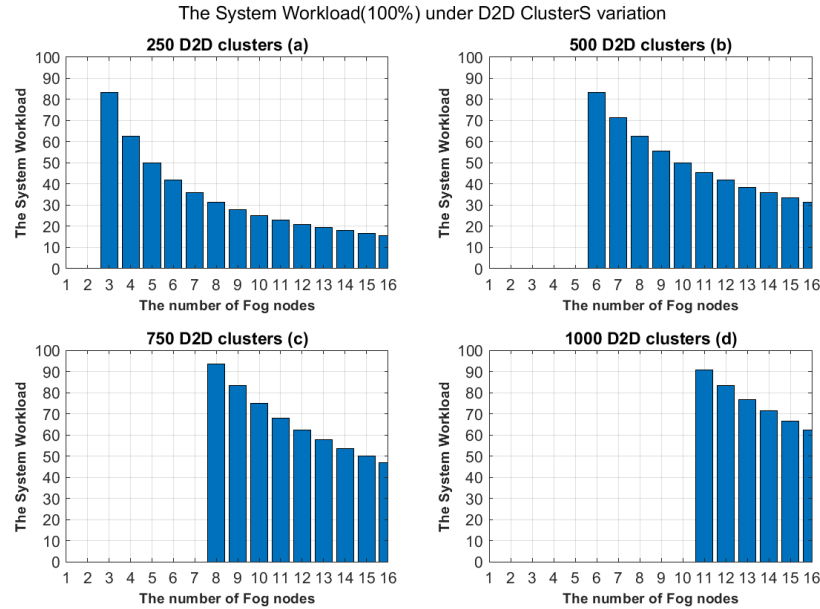


Figura4-5Sarcina de lucru a sistemului cu diferite clustere D2D Densitate (sarcini de serviciu de marime mare)

Astfel, nodurile de ceață 3 până la 4 sunt gata să deservească un grup suplimentar. Mai mult, ca și numărul de clustere D2Da crescut la 500 de clustere (b), volumul de lucru al nodurilor de ceață a ajuns la 60% cu 4 noduri și a scăzut la 40% cu 7 noduri și a continuat să scadă sub 40% pe măsură ce nodurile de ceață au scăzut.

4.1.2 Scenariul de simulare K-Medoids

În această secțiune, este implementat un experiment de simulare pentru desfășurarea/plasarea nodurilor de ceață în locații potențiale (stații de bază). Un set de date de rețea din lumea reală (set de date EUA) este luat în considerare în această lucrare. Setul de date utilizat în această simulare oferit de Universitatea de Tehnologie Swinburne[23]conține coordonatele geografice ale a 125 de stații de bază și 816 utilizatori de telefonie mobilă și ale acelor stații de bază desfășurate în regiunea de afaceri centrală Melbourne.

În acest scenariu, ne concentrăm doar pe locația geografică a stației de bază, iar distribuția utilizatorilor de telefonie mobilă nu face obiectul acestei lucrări. Pentru plasarea nodurilor de ceață, adoptăm abordarea de desfășurare aleatorie a nodurilor de ceață ca o abordare comparativă cu tehnica de grupare K-medoid. numărul de noduri de ceață luate în considerare în acest scenariu a fost determinat din secțiunea anterioară în cadrul sarcinilor de serviciu mari, așa cum este ilustrat în tabelul 5.5.

Masa 4.3 Numărul de noduri de ceață (sarcini de serviciu de dimensiuni mari)

Numărul de cluster D2D	λ_i	Solicitarea totală de servicii (λ)	Numărul de noduri Fog
250	0,5 req/s	125req/s	4 noduri
500	0,5 req/s	250req/s	6 noduri
750	0,5 req/s	375req/s	9 noduri
1000	0,5 req/s	500req/s	12 noduri

Figura 5.8 arată desfășurarea stației de bază după longitudine și latitudine.

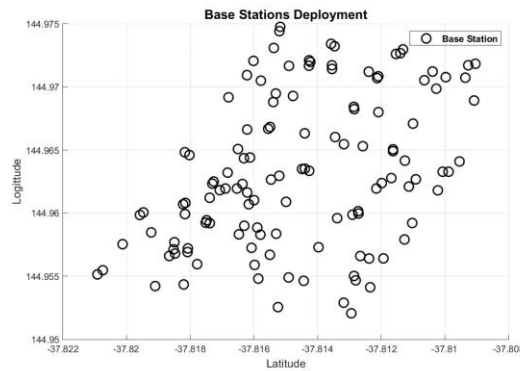
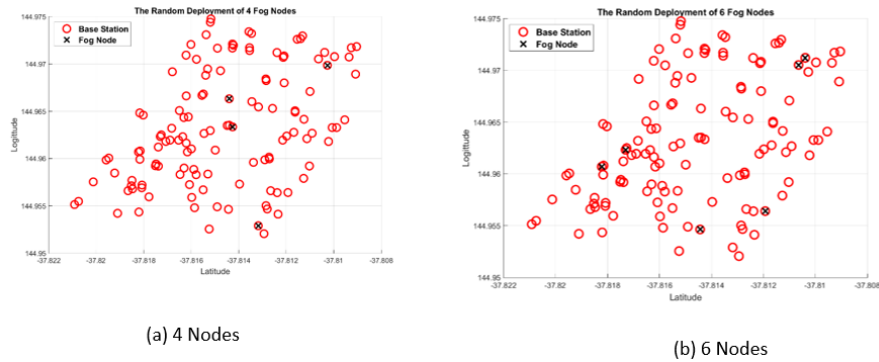


Figura 4-6 Desfășurarea stației de bază

Astfel, nodurile de ceață ar trebui să fie desfășurate astfel încât un set de stații de bază să fie acoperite de un nod de ceață, iar nodul de ceață să fie instalat pe stația de bază care este situată aproape central în acel set.



(a) 4 Nodes

(b) 6 Nodes

Figura 4-7 Implementarea aleatorie a nodului de ceață (4 și 6 noduri)

Abordarea cu desfășurare aleatorie este ilustrată în figurile 5.9 și 5.10; nodurile de ceață nu sunt instalate (distribuite) în mod egal în rețea.

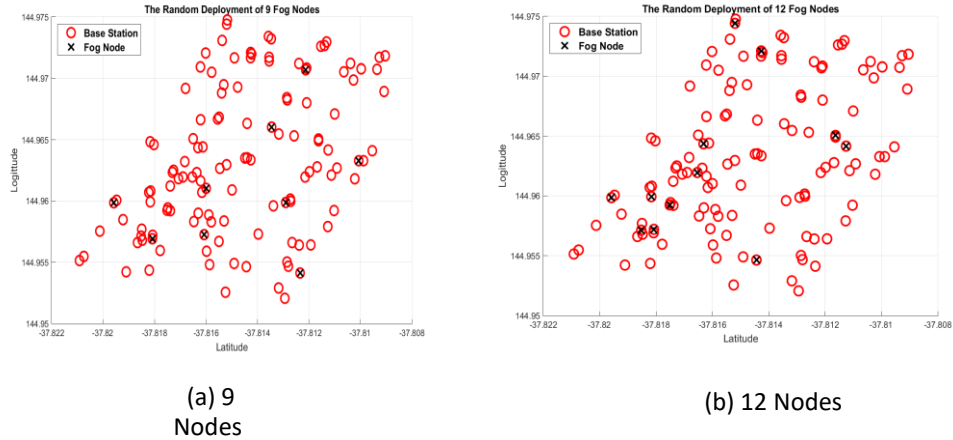


Figura4-8 Implementarea aleatorie a nodului de ceață (9 și 12 noduri)

După cum putem observa, unele noduri de ceață s-au implementat aproape unele de altele, lăsând numărul de stații de bază fără acoperire a nodurilor de ceață.

Astfel, o distanță mare între nodurile de ceață și stațiile de bază. Tehnicile de grupare a K-medoidelor ilustrate în Figura 5.11; setul de stație de bază este împărțit în K set (aici $K = 12$). Stația de bază situată aproape central în fiecare cluster va fi un candidat potrivit pentru instalarea nodului de ceață.

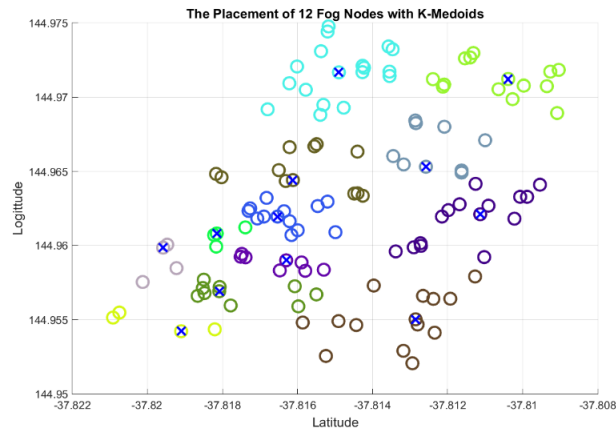


Figura4-9 Plasarea nodurilor de ceață K-Medoids (12 noduri de ceață)

Distribuția nodurilor de ceață a acoperit stația de bază în mod egal; distanța medie dintre nodurile Fog și stațiile de bază de pe cluster este minimă în comparație cu desfășurarea aleatorie ilustrată în Figura 5.12 (b).

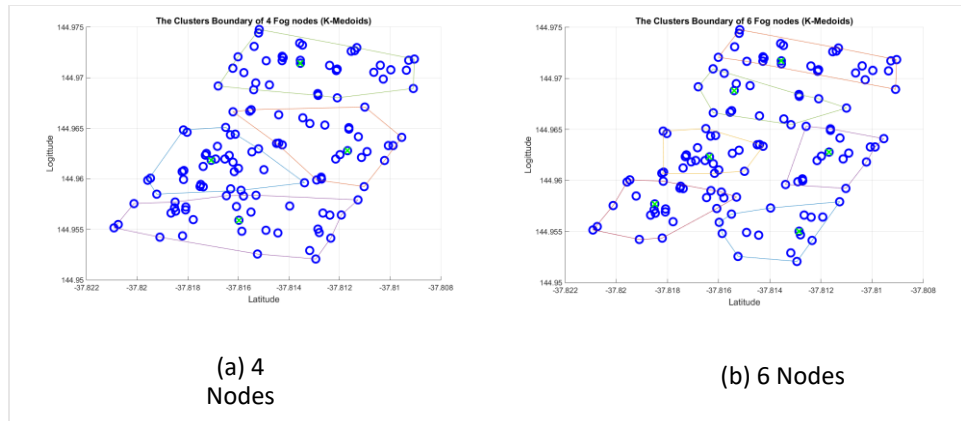


Figura4-10 Limita clusterelor pentru un număr diferit de noduri de ceață (4 și 6 noduri)

Figurile 5.12 și 5.13 au ilustrat definiția clusterului pentru un număr diferit de noduri de ceață. Putem observa, pe măsură ce numărul de cluster crește (adică, creșterea nodurilor de ceață), distanța dintre stațiile de bază și nodurile de ceață scade.

Prin urmare, întârzierea de acces între utilizatorii de telefonie mobilă și nodurile de ceață s-a îmbunătățit. Cu toate acestea, instalarea mai multor noduri de ceață vine cu costul de implementare.

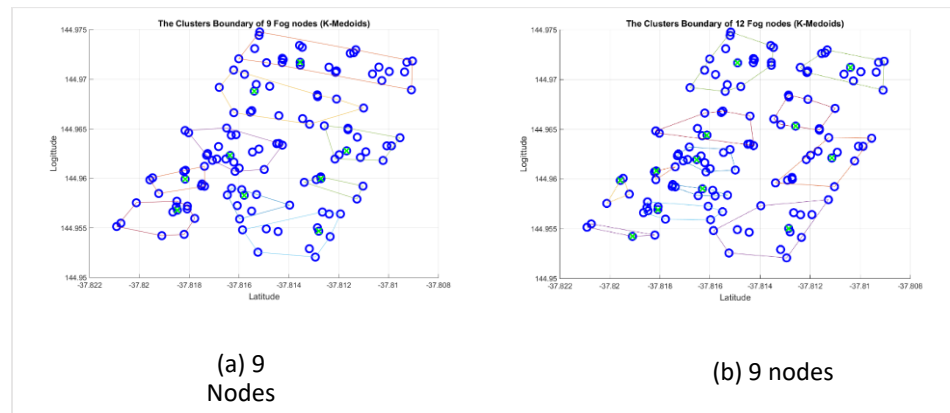


Figura4-11 Limita clusterelor pentru un număr diferit de noduri de ceață (9 și 12 noduri)

4.1.3 Concluzie

În această lucrare, am studiat prin experimente de simulare impactul numărului de noduri de ceață asupra performanței rețelei. Am observat că creșterea numărului de noduri de ceață va reduce timpul de răspuns al solicitărilor de servicii, ceea ce, la rândul său, îmbunătățește latența și QoS din partea utilizatorului. După cum am observat din Figura4-1, și Figura4-2, sistemul va fi stabil după ce vor fi implementate un anumit număr de noduri de ceață; 5 noduri în cazul = 250, 10 noduri pentru = 500 și așa mai departe. $\lambda\lambda$

După atingerea stabilității, numărul de noduri de ceață poate fi crescut pentru a atinge cerințele QoS, așa cum este ilustrat în figura 4-3, și **Error! Reference source not found.**, când trebuie să implementăm aproximativ 4 noduri de ceață pentru a satisface constrângerea QoS (). Același caz pentru , nodurile de ceață instalate în rețea ar trebui să fie mai mari sau egale cu 9 noduri. $\lambda = 250T_{QoS}\lambda = 375$

Rezultatele simulării arată că, volumul de lucru a crescut pe măsură ce numărul clusterelor D2D a crescut, apoi adăugând mai multe noduri de ceață în rețea, se obține o scădere a sarcinii de lucru pe server. De exemplu, când (în $\lambda = 250$ figura 4-4), cu 5 noduri de ceață și mai sus, nodurile de ceață funcționează sub 60%. Astfel, este necesară o echilibrare eficientă a sarcinii pentru a proiecta pentru optimizarea performanței rețelei.

Pentru plasarea nodurilor de ceață, am observat că tehnicile de desfășurare aleatorii nu asigură o distribuție egală a sarcinii de lucru între nodurile de ceață, așa cum se arată în figura 4-7, și figura 4-8. Adică, un grup de stații de bază se vor conecta la nodurile de ceață situate departe de pozițiile lor. Astfel, o latență ridicată rezultată (adică, întârzierea accesului) afectează performanța rețelei. După cum am observat din figura 4-9, stațiile de bază au fost împărțite în K cluster, iar nodurile de ceață instalate la stația de bază sunt situate aproape în centrul aceluși cluster. Mai mult, pe măsură ce numărul de cluster crește, distanța dintre nodurile de ceață și stațiile de bază scade, așa cum se arată în figura 4-10, și figura 4-11.

4.2 Echilibrarea sarcinii SDN pentru rețelele de noduri de ceață

Scenariile de simulare au presupus că există un număr de cluster D2D care generează sarcini de serviciu și acestea sunt considerate eterogene în ceea ce privește lungimea (adică, dimensiunea) și cerințele de calcul de procesare. Prin urmare, sunt necesari timpi de execuție diferiți pentru sarcinile de service.

După cum sa menționat, fiecare nod de ceață rulează un număr de (egal cu 15, 20, 25, 30) și fiecare are anumite specificații, inclusiv puterea de calcul diferită (1000 și 2000), numărul de elemente de procesare, dimensiunea memoriei și altele. În plus, numărul de cluster D2D din rețea a variat de la 100 la 1000 de cluster, iar fiecare cluster generează două tipuri de servicii care sunt diferite ca mărime (500-2500 și 3000-5000 de instrucțiuni pe secundă). Tabelul 5.6 a ilustrat caracteristicile entităților utilizate în această simulare. *VMsVMsVMMIPS*

Masa 4.4 Descrierea parametrilor

Parametrii	Descriere
------------	-----------

<i>clustere D2D</i>	Numărul de clustere D2D	Între 100-1000
	Durata sarcinii (instrucțiuni pe milisecundă)	500-2500 (sarcini de dimensiuni mici) 3000-5000 (sarcini de dimensiuni mari)
<i>Mașini virtuale</i>	Putere de procesare (<i>MIPS</i>)	1000 și 2000
	Dimensiunea memoriei (RAM)	512 MB
	Numărul de nuclee	1
	Un numar de <i>VMs</i>	15, 20, 25, 30 <i>VMs</i>

În această lucrare, CloudSim[24] va fi utilizat pentru modelarea scenariului de simulare de mai sus. această lucrare ia în considerare trei tehnici First Come First Serve (FCFS) care au fost utilizate în comparație cu algoritmul GA în ceea ce privește următoarele valori de performanță:

1. Makespan (M): Makespan este definit ca timpul total de finalizare necesar pentru a executa toate sarcinile din sistem. Valoarea mai mică a makespan a indicat eficiența algoritmilor sau tehnicilor. Se determină cu ecuația 30 *MSMS*
2. Utilizarea resurselor (RU): considerată drept criteriu de dorință pentru furnizorii de servicii și măsoară utilizarea resurselor (levierul resurselor disponibile). Prin urmare, cea mai mare valoare este de dorit. Se calculează așa cum se arată în ecuația 33: *RURU*

$$RU = \frac{\text{Total completion Time of all Tasks}}{\text{makespan}} \quad (4.1)$$

3. Grad de dezechilibru (DI): măsurat cantitatea de distribuție a sarcinii între noduri (VM-uri în cazul nostru). Valoarea mică arată că sarcina sistemului este mai echilibrată. Se determină astfel: *DI*

$$DI = \frac{T_{max} - T_{min}}{T_{avg}} \quad (4.2)$$

Unde T_{max} și T_{min} a prezentat timpul maxim și minim de execuție dintre toate VM-urile și a indicat timpul total de execuție mediu. T_{min} T_{avg}

Algoritmul primul venit, primul servit (FCFS), este un algoritm non-preemptiv. Sarcina de serviciu care vine mai întâi va fi finalizată mai întâi execuția sa și apoi celălalt proces va trebui să aștepte în coadă până când execuția este finalizată[25].

Sunt necesari mai mulți parametri pentru a se stabili pentru funcționarea algoritmului genetic, așa cum este ilustrat în Tabelul 5.7. În principal, rata de încrucișare (C_R) setat la 0,95 și probabilitatea de mutație (M_p) la 0,6.

Masa 4.5 Parametrii algoritmului genetic

Parametrii genetici	Descriere
Dimensiunea populației	50
Metoda de selecție	Ruleta-roata
Rata de încrucișare	0,95
Tip crossover	Două puncte
Rata de mutație	0,6

În această lucrare, trei scenarii de simulare sunt luate în considerare după cum urmează:

- ❖ Primul scenariu a luat în considerare un număr fix de (aproximativ 15) și un număr diferit de clustere D2D care generează sarcini de serviciu între 500-2500 și 5000-3000 de instrucțiuni. În acest scenariu, metrica de performanță menționată mai devreme va măsura și analiza între algoritmi de mai sus. $VMsVMs$
- ❖ În al doilea scenariu, numărul de sarcini de serviciu va fi fixat la un număr maxim de sarcini primite (aproximativ 1000 de sarcini), unde numărul de sarcini a variat de la 15 la 30. În continuare, și, respectiv, vor fi analizate. $VMVMsakespanRUDI$
- ❖ Al treilea scenariu ia în considerare două tipuri de (Scăzut și ridicat) care sunt diferite ca putere de procesare. Lowul are o putere de procesare de aproximativ 1000, iar cel mare are o putere de procesare de aproximativ 2000. În plus, numărul a variat de la 10 la 30. $VMsVMsVMsMIPS VMsMIPSVMs$

4.2.1 Primul scenariu (diferite dimensiuni ale sarcinilor de serviciu)

După cum sa menționat mai devreme, numărul de clustere D2D distribuite în rețea între 100 și 1000 de clustere, iar aceste clustere au generat sarcini de serviciu în două tipuri de dimensiuni mici (500-2500) și mari (3000-5000) și un număr de în sistem fixat la 15. VMs

Figura 5.16 a arătat performanța a două tehnici de echilibrare a sarcinii sub un număr diferit de clustere D2D; tehnica FCFS oferă o valoare makespan aproape similară

cu tehnica GA la un număr redus de clustere. Acest caz ar putea fi observat atunci când numărul de clustere D2D este între 100 și 300. Cu toate acestea, pe măsură ce numărul de clustere a crescut, FCFS oferă un makespan mare în comparație cu algoritmul GA.

De exemplu, când avem aproximativ 600 de clustere D2D, FCFS oferă valoarea makespan-ului peste 0,05, iar în cazul GA, valoarea makespan-ului este sub 0,05. Mai mult, numărul clusterelor D2D a ajuns la 900 până la 1000; GA a avut rezultate mai bune decât FCFS.

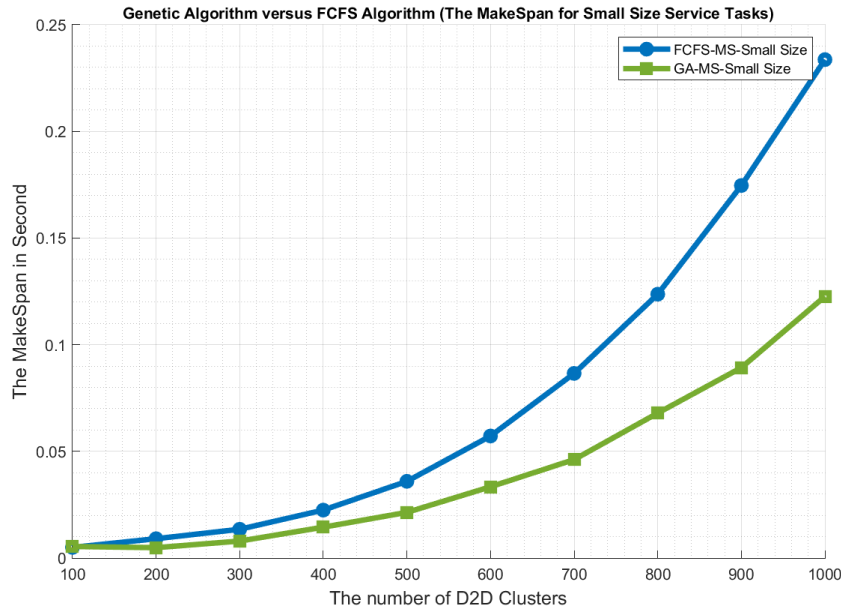


Figura4-12Makespan of GA vs. FCFS (sarcini de service de dimensiuni mici)

Următoarea figură a arătat cazul când clusterelor D2D au generat sarcini de serviciu de dimensiuni mari. După cum sa observat, GA oferă o valoare de makespan scăzută în comparație cu tehnica FCFS.

unde FCFS a suferit din cauza valorii de makespan ridicate pe măsură ce numărul clusterelor a crescut în rețea.

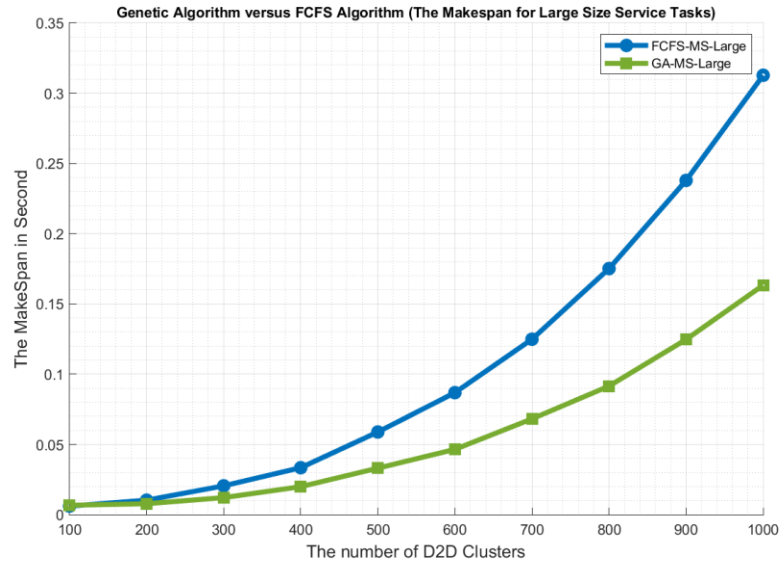


Figura4-13Makespan of GA vs. FCFS (sarcini de service de dimensiuni mari)

Figura 5.18 a ilustrat performanța generală a algoritmilor GA și FCSF în sarcini de serviciu de dimensiuni mici și mari.

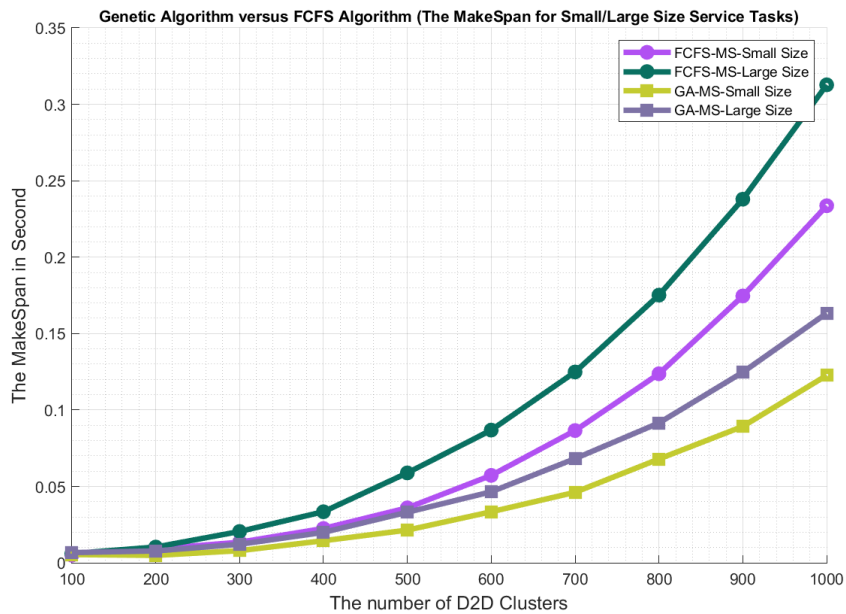


Figura4-14The Makespan pentru sarcini de service de dimensiuni mici/mari

Gradul de dezechilibru (DI) pentru sarcinile de service de dimensiuni mici este prezentat în Figura 5.19. după cum sa menționat mai devreme, gradul de dezechilibru a măsurat cantitatea de VM-uri de distribuție a sarcinii în cazul nostru. Cea mai mică valoare a arătat că VM-urile au primit o cantitate aproape egală de încărcare.

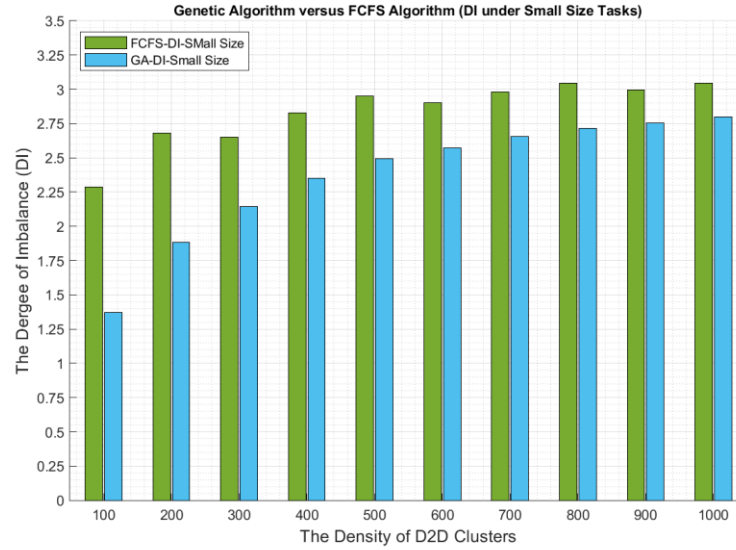


Figura4-15 Gradul de dezechilibru (sarcini de dimensiuni mici)

Algoritmul GA oferă o valoare scăzută a DI în comparație cu FCFS, așa cum se arată în figura de mai sus, cu un număr diferit de clustere D2D.

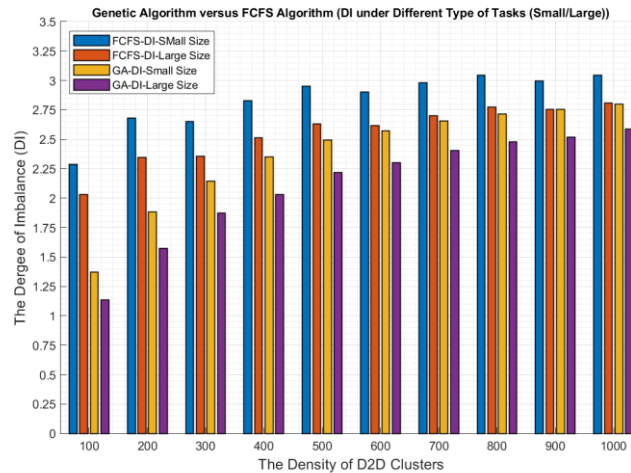


Figura4-16 Gradul de dezechilibru (diferite tipuri de sarcini de serviciu)

Figura 5.20 a ilustrat performanța GA și FCFS a sarcinilor de serviciu de dimensiuni mici și mari cu un număr diferit de clustere D2D. De exemplu, când numărul de clustere din rețea este egal cu 500, DI cu GA are o valoare sub 2,5, iar pe de altă parte, valoarea DI cu FCFS este de aproximativ 3,5. În același caz, când clusterele D2D sunt egale cu 1000, valoarea DI este egală cu aproximativ 2,75 (GA) și aproape 3,0 pentru FCFS.

4.2.2 Al doilea scenariu (implementarea diferitelor VM)

În acest scenariu, numărul de clustere D2D din rețea va fi fixat la 1000 de clustere unde numărul de VM-uri implementate a variat de la 15 la 30. Mai mult, două tipuri de sarcini de serviciu (mici și mari) sunt luate în considerare în această secțiune.

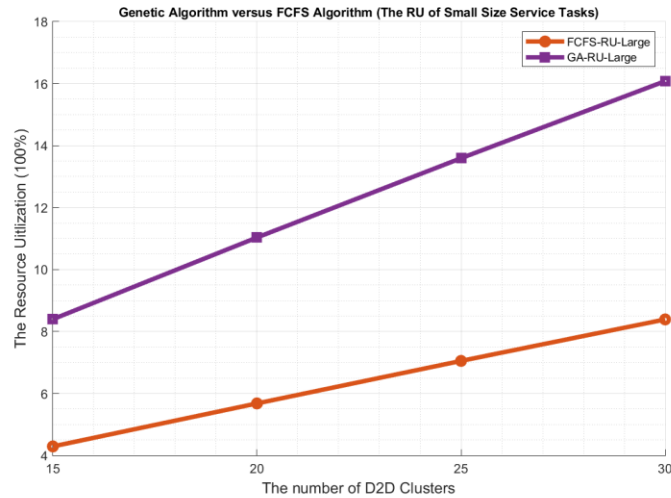


Figura4-17RU a GA vs. FCFS (Small Size Service Tasks)

După cum sa menționat, cea mai mare valoare a utilizării resurselor (RU) este de dorit. Cu toate acestea, FCFS oferă cea mai mică valoare a RU în comparație cu GA.

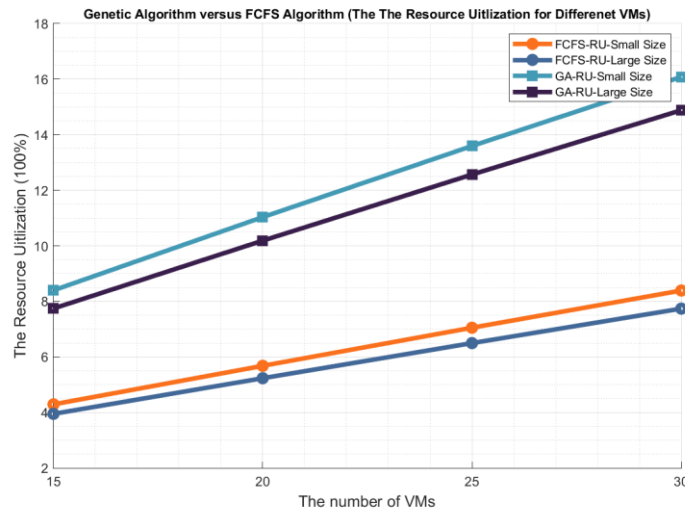


Figura4-18Utilizarea resurselor pentru diferite VM (sarcini de serviciu de dimensiuni mici și mari)

Prin urmare, pe măsură ce a crescut numărul de VM; AG gestionează eficient resursa. De exemplu, în cazul în care numărul de VM este egal cu 20; Valoarea RU este egală cu 13%; pe de altă parte, FCFS a utilizat resursa aproximativ 5% (vezi Figura 5.21).

Figura 5.22 arată utilizarea resurselor cu diferite tipuri de sarcini de serviciu. După cum observăm, pe măsură ce crește numărul de VM, GA funcționează mai bine decât FCFS.

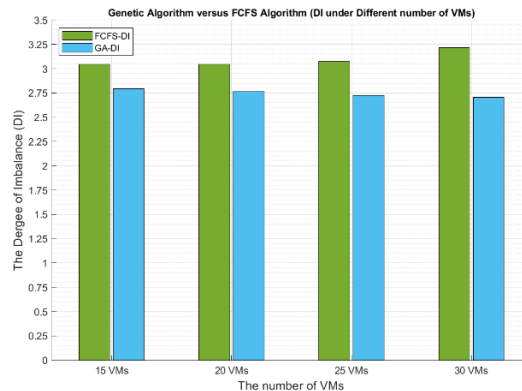


Figura4-19Gradul de dezechilibru (număr diferit de VM)

Cu toate acestea, GA cu sarcini de service de dimensiuni mari oferă o valoare scăzută în comparație cu sarcinile de service de dimensiuni mici; Motivul este că GA oferă o durată mare, așa cum se arată în figura 5.22. Același caz pentru algoritmul FCFS.

Figura 5.23 a ilustrat performanța GA în diferite VM-uri în termenul Gradului de dezechilibru.

GA oferă o valoare scăzută a DI cu o valoare ușor diferită pe măsură ce crește numărul de VM. De exemplu, la 15 VM, valoarea DI este de aproximativ 2,76, iar la 30 de VM, aproximativ 2,6.

4.2.3 Al treilea scenariu (VM-uri scăzute și ridicate)

În acest scenariu, au fost luate în considerare două tipuri de VM (VM-uri scăzute cu putere de calcul de aproximativ 1000 și VM-uri mari cu 2000), iar numărul de VM-uri a variat de la 10 la 30. De asemenea, densitatea clusterelor D2D va fi fixată la 1000 de cluster. care a generat sarcini de serviciu mici și mari.*MIPSMIPS*

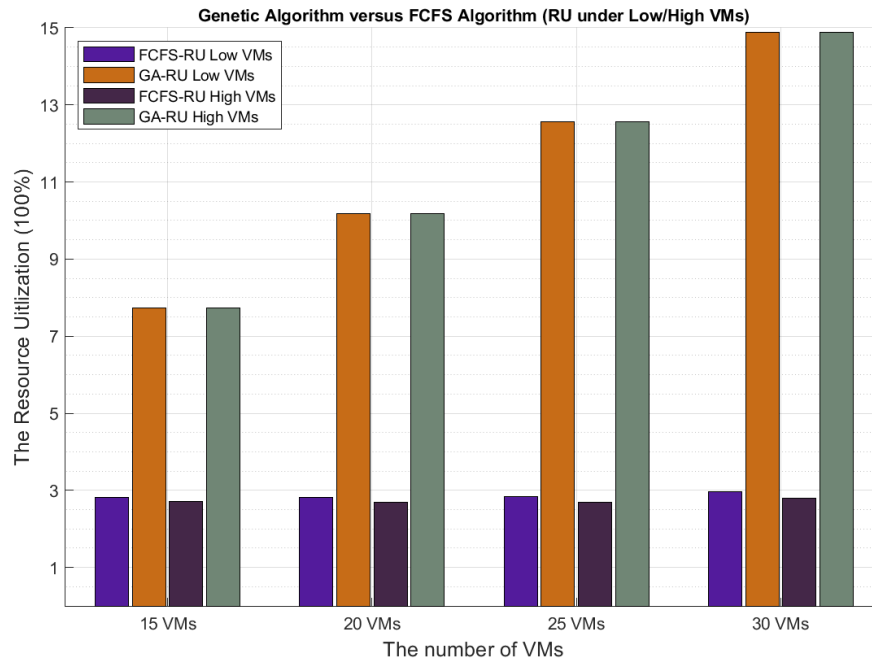


Figura4-20 Utilizarea resurselor pentru VM-uri scăzute/înalte

După cum este ilustrat în Figura 5.24, FCFS suferă de o utilizare scăzută a resurselor în comparație cu algoritmul GA. De exemplu, când numărul de VM-uri implementate este egal cu 15, RU la FCFS sub 3%, iar pe de altă parte, GA oferă o utilizare mare a resurselor, egală cu aproximativ 7%.

Performanța algoritmilor FCFS și GA în ceea ce privește distribuția sarcinii este ilustrată în Figura 5.25. După cum se observă, numărul crescut de VM nu are niciun efect asupra performanței FCFS pentru a ajunge la o valoare DI similară cu cea furnizată de GA.

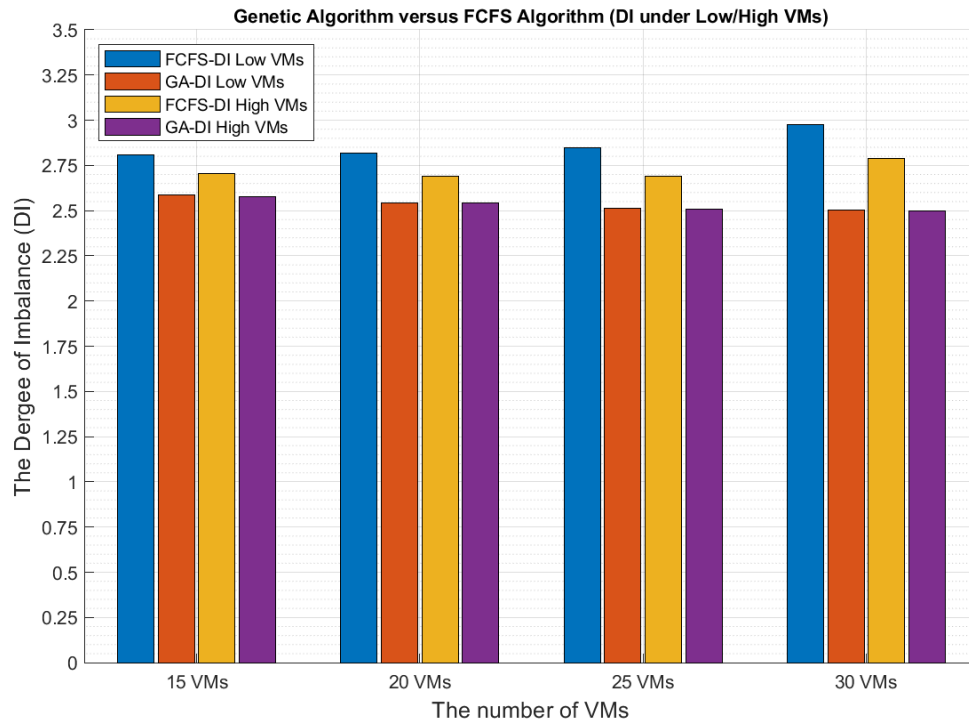


Figura4-21 Gradul de dezechilibru (VM-uri scăzute/înalte)

4.2.4 Concluzie

În această secțiune, problemele de echilibrare a sarcinii în calculul de ceață au fost studiate și modelate matematic. Problema echilibrării sarcinii este considerată o problemă NP-Hard. În consecință, algoritmul genetic a fost considerat principala tehnică de rezolvare a echilibrării sarcinii. Experimentele de simulare prezentate, algoritmul genetic a funcționat mai bine decât alți algoritmi, și anume, FCFS în ceea ce privește makespan-ul, utilizarea resurselor și factorul de dezechilibru.

Am observat că algoritmul genetic oferă o valoare mai bună a makespan-ului decât FCFS și este prezentat în figuri Figura4-12, Figura4-13, și Figura4-14. După cum este ilustrat, de exemplu, în figura Figura4-12 când avem aproximativ 600 de clustere D2D, FCFS dă valoarea makespan-ului peste 0,05, iar în cazul GA, valoarea makespan-ului este sub 0,05.

Rezultatele simulării arată că algoritmul genetic oferă un grad scăzut de dezechilibru (valoarea scăzută este de dorit) aproximativ 2,4 la 500 de clustere unde FCFS are 3,0 DI pentru sarcini de servicii de dimensiuni mici, așa cum se arată în figura Figura4-15. FCFS funcționează cel mai rău în comparație cu algoritmul genetic, așa cum se vede în figura Figura4-16 în ambele cazuri de tipuri de sarcini de serviciu.

Al doilea scenariu a ilustrat cazul unui număr diferit de VM instalate în rețea și cantitatea maximă de sarcini de serviciu. Pe măsură ce numărul de VM a crescut, algoritmul GA oferă o resursă mare, egală cu 13%, iar când numărul de VM este egal cu 20, iar pe de altă parte, FCFS suferă de cea mai scăzută resursă utilizată (FiguraFigura4-19). Performanța ambelor tehnici în funcție de diferite sarcini de serviciu arată că GA are cea mai mare utilizare a resurselor, așa cum se vede în figuraFigura4-18; la 20 de VM, GA oferă o valoare de RU de aproximativ 10% (dimensiune mică) și 12 (dimensiune mare).

Al treilea scenariu a luat în considerare puterea diferită de procesare a VM-urilor. Algoritmul genetic (GA) are o utilizare mare a resurselor în ambele cazuri (VM-uri scăzute și mari), așa cum se vede în figuraFigura4-20, când numărul de VM-uri implementate este egal cu 15 RU la FCFS sub 3%. Pe de altă parte, GA oferă o utilizare ridicată a resurselor, egală cu aproximativ 7%. După cum se observă în figuraFigura4-21, numărul crescut de VM nu are nici un efect asupra atingerii FCFS la o valoare DI similară din GA.

Capitolul 6

5 Discuții și lucrări viitoare

În această lucrare, a fost investigată integrarea calculului de ceață în comunicarea D2D. Prin urmare, principala contribuție a acestei lucrări este determinarea numărului necesar de noduri de ceață pentru a furniza un serviciu de ceață utilizatorilor D2D și locațiile lor candidate în rețea, distribuind, de asemenea, sarcina de lucru în mod egal între nodurile de ceață. Calculul de ceață este implementat ca noduri de calcul care furnizează serviciul de ceață clusterelor D2D.

Această lucrare a folosit modelul de coadă multi-server pentru a determina numărul necesar de noduri de ceață în rețea. În plus, serverele cloud de la distanță au fost, de asemenea, modelate ca modele de cozi cu mai multe servere. Mai mult, desfășurarea nodurilor de ceață a fost implementată conform algoritmilor de clustering, și anume K-medoids. Mai mult decât atât, problema echilibrării sarcinii a fost formulată cu timpul de așteptare pentru completarea matricei (ETC). Cu implicarea SDN, algoritmul genetic a fost considerat principala abordare pentru rezolvarea problemei de echilibrare a sarcinii. O sugestie pentru lucrări viitoare poate fi rezumată după cum urmează:

1. Luând în considerare mobilitatea utilizatorilor, menținând în același timp conectivitatea serviciilor care s-a descărcat în rețeaua Fog Nodes. Mai mult, aceste servicii ar putea avea caracteristici diferite, precum latența.
2. Luând în considerare orchestrarea SDN a diferitelor secțiuni (de exemplu, secțiunea de rețea vehiculară și secțiunea de rețea mobilă). SDN Orchestration menține și gestionează serviciile furnizate de diferite slice.

Lista rapoartelor științifice

Această listă cuprinde doar rapoarte de cercetare program de doctorat.

1. Raport Științific Nr.1, Comunicații Device to Device în tehnologii 4G și 5G, Universitatea Politehnică București, Școala Doctorală de Electronică, Telecomunicații și Tehnologia Informației, SD-ETTI-B, Contract de studii universitare, Doctorat, martie 2017.
2. Raport Științific Nr.2, Managementul Resurselor și Controlul în D2D pe baza SDN/NFV, Universitatea Politehnică din București, Școala Doctorală de Electronică, Telecomunicații și Tehnologia Informației, SD-ETTI-B, Contract de studii universitare, Doctorat, noiembrie 2017.
3. Raportul științific nr. 3, *Rutare Multi-Hop centralizată pentru comunicare D2D*, Universitatea Politehnică din București, Școala Doctorală de Electronică, Telecomunicații și Tehnologia Informației, mai-2019.
4. Raportul științific nr. 4, *Fog Computing și integrarea rețelelor D2D*, Universitatea Politehnică din București, Școala Doctorală de Electronică, Telecomunicații și Tehnologia Informației, martie 2021.

Rezumatul contribuțiilor originale

Rezumatul principalele contribuții ale acestei teze sunt:

1. Modelarea calculului de ceață ca o rețea de rețea de noduri de ceață (FCN) cu teoria Queuing.
2. Numărul optim de noduri de ceață a fost determinat printr-o serie de ecuații formulate sub constrângeri QoS.
3. Rezultatele simulării prezentate ca creșterea numărului de noduri de ceață vor reduce timpul de răspuns la cererea de serviciu și vor satisface pragul QoS.
4. Rezultatele simulării arată că sarcina de lucru a crescut pe măsură ce numărul clusterelor D2D a crescut. Mai mult, pe măsură ce se adaugă mai multe noduri de ceață în rețea, volumul de lucru a scăzut pentru fiecare nod de ceață.
5. Tehnica de grupare K-Medoids adoptată pentru implementarea nodurilor de ceață cu asigurarea distanței minime între stațiile de bază și nodurile de ceață.

6. Stația de bază este stabilită în rețea împărțită în clustere de serval, iar în fiecare cluster vor fi instalate noduri Fog aproape în centrul acestor clustere.
7. Rezultatele simulării au arătat că pe măsură ce numărul de clustere crește, distanța dintre nodurile de ceață și stațiile de bază scade. Cu alte cuvinte, distanța dintre nodul de ceață și stația de bază este redusă pe măsură ce crește numărul de clustere.
8. Rezultatele simulării au arătat că integrarea SDN la calculul de ceață a arătat cea mai mică valoare a timpului de răspuns în comparație cu serverele de la distanță din cloud.
9. O caracteristică diferită a nodului de ceață luată în considerare, inclusiv numărul variat de mașini virtuale, puterea de procesare scăzută și mare a mașinilor virtuale.
10. Rezultatele simulării prezentate, algoritmul genetic a funcționat mai bine decât algoritmul FCFS în ceea ce privește makespan-ul, utilizarea resurselor și factorul de dezechilibru.

Publicații

Mustafa Khaleel Hamadani, și Eugen Borcoci. „Plasarea nodurilor de ceață pentru rețele D2D”. În 2021, a 44-a Conferință internațională privind telecomunicațiile și procesarea semnalelor (TSP), pp. 152-156. IEEE, 2021. [Acceptat, invitat să fie publicat într-o formă extinsă pentru Sensors Journal]. Link: <https://ieeexplore.ieee.org/document/9522637>

Mustafa Khaleel Hamadani, și Eugen Borcoci. „Fog Computing și integrarea rețelelor D2D”. În 2021, IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 1-6. IEEE, 2021.[Admis]. Link: <https://ieeexplore.ieee.org/document/9527740>

Mustafa Khaleel Hamadani, Mahdi AA. Rutare multi-hop centralizată pentru comunicarea de la dispozitiv la dispozitiv: simulare și rezultate. În 2019, a 11-a Conferință internațională privind electronică, computere și inteligență artificială (ECAI) 27 iunie 2019 (p. 1-6). IEEE. [Admis]. Link: <https://ieeexplore.ieee.org/document/9042024/>

Mustafa Khaleel Hamadani, Al-Alwash Husam Mahdi, Tehnici centralizate de rutare multisop pentru comunicare de la dispozitiv la dispozitiv, ICN 2019: The Eighteenth International Conference on Networks, ARIA, 2019. ISBN: 978-1-61208-695-8. [Admis]. Link: http://www.thinkmind.org/index.php?view=article&articleid=icn_2019_1_30_30040

Al-Alwash Husam Mahdi, Mustafa Khaleel Hamadani, Vehicular to Grid echnologies– A Survey on Architectures and Solutions, ICN 2019: The Eighteenth International Conference on Networks, ARIA, 2019. ISBN: 978-1-61208-695-8. [Admis]. Link: http://www.thinkmind.org/index.php?view=article&articleid=icn_2019_2_20_30045

Silviu-Andrei Lazar, Mustafa Khaleel Hamadani, Carmen-Elena Stefan, Optimization Analysis of VANET's Control Plane for Safety Application Traffic, 2018 International Conference on Communications (COMM). [Admis]. Link: <https://ieeexplore.ieee.org/document/8484272/>

Mustafa Khaleel Hamadani, și Eugen Borcoci. " Load Balancing Techniques for Fog Computing integrate în rețelele D2D.". [trimis la **Buletin Științific - Universitatea POLITEHNICA din București**].

Referință

- [1] Seria M., „Viziunea IMT – Cadrul și obiectivele generale ale dezvoltării viitoare a IMT pentru 2020 și ulterior”, *Recomandarea ITU*, vol. 2083, p. 21, 2015.
- [2] G. Fodoret *al.*, „O prezentare generală a componentelor tehnologiei de comunicații de la dispozitiv la dispozitiv în METIS”, *Ieee Access*, vol. 4, p. 3288–3299, 2016.
- [3] G. Araniti, A. Raschella, A. Orsino, L. Militano și M. Condoluci, „Device-to-device communications over 5G systems: Standardization, challenges and open issues”, în *5G mobile communications*, Springer, 2017, pp. 337–360.
- [4] RI Ansari *et al.*, „Rețele 5G D2D: tehnici, provocări și perspective de viitor”, *IEEE Systems Journal*, vol. 12, nr. 4, p. 3970–3984, 2017.
- [5] X. Lin, JG Andrews, A. Ghosh și R. Ratasuk, „O privire de ansamblu asupra serviciilor de proximitate dispozitiv-la-dispozitiv 3GPP”, *IEEE Communications Magazine*, vol. 52, nr. 4, p. 40–48, 2014.
- [6] O. Hayat, R. Ngah, SZM Hashim, MH Dahri, RF Malik și Y. Rahayu, „Descoperirea dispozitivelor în comunicarea D2D: un sondaj”, *Acces IEEE*, vol. 7, p. 131114–131134, 2019.
- [7] Z. Zhu, X. Luo, K. Wang și Y. Yang, „Integrarea D2D cu Edge/Fog Computing”, *Wiley 5G Ref: Referința 5G esențială online*, pp. 1–18, 2019.
- [8] P. Mell, T. Grance și alții, „The NIST definition of cloud computing”, 2011.
- [9] CM Mohammed, SRMZeebaree și alții, „Suficient comparison between cloud computing services: IaaS, PaaS, and SaaS: A review”, *International Journal of Science and Business*, vol. 5, nr. 2, p. 17–30, 2021.
- [10] V. Prokhorenko și MA Babar, „Arhitectural resilience in cloud, fog and edge systems: a survey”, *Acces IEEE*, vol. 8, p. 28078–28095, 2020.
- [11] M. Chen, Y. Hao, Y. Li, C.-F. Lai și D. Wu, „Despre descărcarea de calcul la cloudlet ad-hoc: moduri de arhitectură și servicii”, *Revista IEEE Communications*, vol. 53, nr. 6, p. 18–24, 2015.

- [12] M.Firdhous, O. Ghazali și S. Hassan, „Fog computing: Will it be the future of cloud computing?”, 2014.
- [13] OCAW Group și alții, „OpenFog reference architecture for fog computing”, OPFRA001, vol. 20817, p. 162, 2017.
- [14] YP Llerena și PRLGondim, „SDN-controller placement for D2D communications”, IEEE Access, vol. 7, p. 169745–169761, 2019.
- [15] YC Hu, M. Patel, D. Sabella, N. Sprecher și V. Young, „Mobile edge computing—A key technology to 5G”, *Cartea albă ETSI*, vol. 11, nr. 11, p. 1–16, 2015.
- [16] R. Roman, J. Lopez și M. Mambo, „Mobile edge computing, fog et al.: Un sondaj și analiză a amenințărilor și provocărilor de securitate”, *Sisteme informatice de generație viitoare*, vol. 78, p. 680–698, 2018.
- [17] YC Hu, M. Patel, D. Sabella, N. Sprecher și V. Young, „Mobile edge computing—A key technology to 5G”, *Cartea albă ETSI*, vol. 11, nr. 11, p. 1–16, 2015.
- [18] Z. Pang, L. Sun, Z. Wang, E. Tian și S. Yang, „A survey of cloudlet based mobile computing”, în *2015 Conferința internațională privind cloud computing și Big Data (CCBD)*, 2015, p. 268–275.
- [19] AVDastjerdi și R. Buyya, „Fog computing: Helping the Internet of Things realize its potential”, *Computer*, vol. 49, nr. 8, p. 112–116, 2016.
- [20] S. Yi, Z. Hao, Z. Qin și Q. Li, „Fog computing: Platform and applications”, în *2015 Al treilea workshop IEEE pe subiecte fierbinți în sistemele și tehnologiile web (HotWeb)*, 2015, p. 73–78.
- [21] P. Habibi, M.Farhoudi, S. Kazemian, S. Khorsandi și A. Leon-Garcia, „Fog computing: a comprehensive architectural survey”, IEEE Access, vol. 8, p. 69105–69133, 2020.
- [22] V. Prokhorenko și MA Babar, „Arhitectural resilience in cloud, fog and edge systems: a survey”, *Acces IEEE*, vol. 8, p. 28078–28095, 2020.
- [23] E.Haleplidis, K. Pentikousis, S. Denazis, JH Salim, D. Meyer și O. Koufopavlou, „Software-defined networking (SDN): Layers and architecture terminology”, RFC 7426, 2015.
- [24] F.Hensh, M. Gupta și MJ Nene, „Mist-Edge-Cloud (MEC) Computing: An Integrated Computing Architecture”, în *2021 a Doua Conferință Internațională pentru Electronică și Sisteme de Comunicații Durabile (ICESC)*, 2021, pp. 1035–1040.
- [25] A. Islam, A. Debnath, M. Ghose și S. Chakraborty, „Un sondaj privind descărcarea sarcinilor în calculul de margine cu acces multiplu”, *Jurnalul de arhitectură a sistemelor*, vol. 118, p. 102225, 2021.