УДК: 004.72

DOI: 10.31673/2412-9070.2025.025489

Д. О. КУЛЬЧИЦЬКИЙ[1], студент;
ORCID: 0009-0009-4060-315X
Є. В. ЖУКОВ[2]**,** ад'юнкт,
ORCID: 0009-0002-1251-946X
[1] Київський національний університет імені Тараса Шевченка
[2] Національний університет оборони України, Київ

МОДЕЛЬ КЕРУВАННЯ ВІРТУАЛІЗОВАНИМИ МЕРЕЖЕВИМИ ФУНКЦІЯМИ ЗА УМОВ ДИНАМІЧНИХ ЗМІН НАВАНТАЖЕННЯ

У статті розглядається проблема підвищення ефективності керування віртуалізованими мережевими функціями (NFV) у середовищі програмно-конфігурованих мереж (SDN) за умов динамічних змін навантаження та вимог до якості обслуговування. Зростання масштабів мереж і ускладнення їхньої структури вимагає впровадження інтелектуальних механізмів автоматизації керування мережевою інфраструктурою. Запропоновано модель інтелектуалізованої NFV, що інтегрує нечітке продукційно-логічне виведення (правила типу «якщо-то») у процес оркестрації віртуальних мережевих функцій. Така модель дозволяє на основі моніторингу мережевих метрик та стану ресурсів у режимі реального часу автоматично приймати рішення щодо масштабування, розподілу ресурсів та маршрутизації, адаптуючи роботу мережі до поточних умов.

Обґрунтовується, що традиційні підходи з жорсткими пороговими налаштуваннями або статичними політиками не здатні забезпечити належну гнучкість і швидкодію у сучасних мережах. Натомість використання нечіткої логіки дає змогу обробляти неточні вхідні дані, враховувати експертні знання і досягати плавного адаптивного керування без різких стрибків. У межах роботи представлено структуру запропонованої моделі, алгоритм її функціонування та приклади нечітких правил. Показано, що застосування даного підходу забезпечує більш ефективне використання ресурсів центру обробки даних, підтримує стабільні значення затримки та інших QoS-показників навіть при різких змінах трафіку, і знижує ризик перевантаження мережі. Це особливо актуально для критичних сервісів, що потребують безперервної роботи та низьких затримок (реального часу – VoIP, відеострімінг, IoT тощо).

Ключові слова: програмно-конфігуровані мережі; віртуалізація мережевих функцій; нечітке продукційно-логічне виведення; оркестрація; якість обслуговування (QoS); інтелектуальна система керування.

Вступ

У сучасних комп'ютерних мережах спостерігається стрімке зростання обсягів трафіку та кількості сервісів, що висуває підвищені вимоги до гнучкості та масштабованості інфраструктури. Технології Software-Defined Networking (SDN) та Network Function Virtualization (NFV) стали ключовими складовими мережевої софтверизації, пропонуючи відокремлення керування від обладнання та перенесення мережевих функцій у програмну площину [1]. Зокрема, SDN архітектурно розділяє площину контролю і площину передавання даних, централізуючи управління мережею. Натомість NFV дозволяє виконувати функції, такі як маршрутизація, брандмауери, балансування навантаження у вигляді програмних модулів (віртуальних мережевих функцій, VNF), що можуть динамічно розгортатися на стандартному апаратному забезпеченні [2]. Ці підходи суттєво підвищують гнучкість і швидкість впровадження нових сервісів, усуваючи необхідність у спеціалізованому обладнанні та прискорюючи інновації у мережевій сфері. Однак поява SDN/NFV породжує і нові виклики.

Централізація управління та віртуалізація функцій вимагають більш інтелектуальних засобів моніторингу та керування, оскільки традиційні методи з фіксованими порогами або вручну заданими політиками не справляються з динамікою сучасних мереж.

Актуальність теми зумовлена потребою забезпечити якість обслуговування (QoS) та надійність мережевих сервісів у реальному часі при постійному коливанні попиту. Наприклад, у пікові години може різко зрости навантаження на мережевий шлюз або сервер застосунків. Якщо використовувати лише жорстко задані пороги завантаження CPU або трафіку для масштабування, це може призводити або до запізнілої реакції (коли поріг занадто високий і система не встигає масштабуватися), або до "хитання" системи (надто чутливі пороги викликають часті перемикання станів). У той же час, активні методи моніторингу (генерація тестового трафіку) дозволяють проактивно виявляти деградацію, але створюють додаткове навантаження на мережу та можуть не відображати реального трафіку. Пасивні методи, навпаки, аналізують справжні дані користувачів, проте не дають можливості передбачити проблеми, поки ті не проявляться. Таким чином, виникає потреба в об'єднанні декількох підходів та застосуванні інтелектуальних алгоритмів для прийняття рішень у режимі, близькому до реального часу.

Постановка проблеми

Проблема, на розв'язання якої спрямовано дане дослідження, полягає у створенні адаптивної системи керування віртуалізованими мережевими функціями, здатної автоматично і оперативно реагувати на зміну мережевих умов (навантаження, стан каналів, вимоги додатків) без втручання адміністратора. На практиці це означає необхідність прийняття рішень про виділення додаткових ресурсів для певного VNF, міграцію функцій на інші вузли, переналаштування маршрутизації трафіку або зміну пріоритетів обслуговування – і все це в декілька разів швидше, ніж це може зробити людина.

Аналіз останніх досліджень і публікацій

Традиційні засоби оркестрації NFV часто використовують заздалегідь визначені політики масштабування (наприклад, розгорнути додатковий екземпляр мережевої функції, коли використання CPU перевищує 80%). Але такий підхід не враховує комплексну ситуацію: перевищення порогу завантаження може бути тимчасовим сплеском або спричинене зовнішніми факторами і негайне масштабування не завжди оптимальне [1-4]. З іншого боку, надто консервативні пороги призводять до деградації якості, перш, ніж система відреагує. Проблема ускладнюється тим, що в умовах мереж з програмним керуванням можливе часте перепланування маршрутів та перерозподіл трафіку, що може як розвантажити, так і перевантажити окремі VNFs. Тому статичні правила не забезпечують достатньої ефективності [5,6]. Отже, необхідно розробити підхід, який би враховував невизначеність і нечіткість характерних для мережевих метрик ситуацій. Нечітка логіка надає апарат для формалізації таких понять і прийняття рішень на їх основі. Відтак, постає науково-прикладна задача розроблення моделі системи керування NFV, що використовує нечітке продукційно-логічне виведення для динамічної оркестрації мережевих функцій у SDN середовищі.

Виклад основного матеріалу

Прикладом використання моделей, методів та технологій інтелектуалізації у комп'ютерних мережах є так звана інтелектуально-конфігурована мережа (KDN – Knowledge-Defined Networking [7]), яку в деякій україномовній науковій літературі називають мережею, конфігурованою на знаннях [8]. Як відомо, традиційна мережа базується на апаратному забезпеченні, оскільки площина керування поєднується з площиною даних. Програмно-конфігурована мережа (SDN – Software-Defined Networking) [7] має логічно централізовану площину керування для підвищення програмованості та гнучкості мереж. Інтелектуально-конфігурована мережа (KDN – Knowledge-Defined Networking) є вдосконаленою SDN, в якої відокремлена площина керування від логіки керування та є нова площина – площина знань (рис. 1).

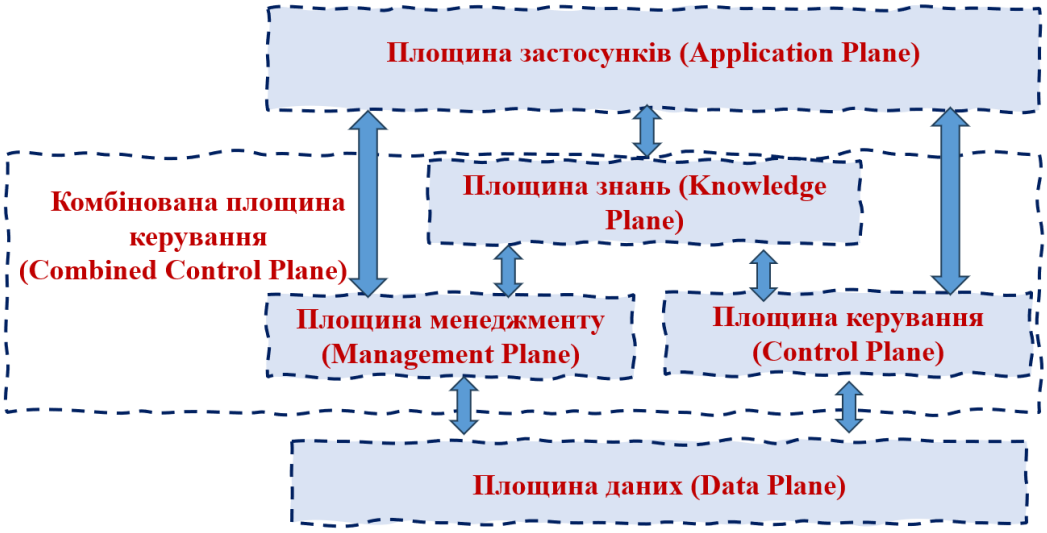

Рис. 1. Схема архітектури KDN

Площина знань – це база знань та механізм виведення нових знань, яка відокремлена від логіки керування та призначена для генерування знань на основі даних, зібраних із мережі. KDN – це архітектура наступного покоління для мереж, що самонавчаються, самоорганізуються та саморозвиваються, з високим рівнем автоматизації та інтелекту.

На рис. 2 надана власна модель взаємодій моделі OSI та KDM.

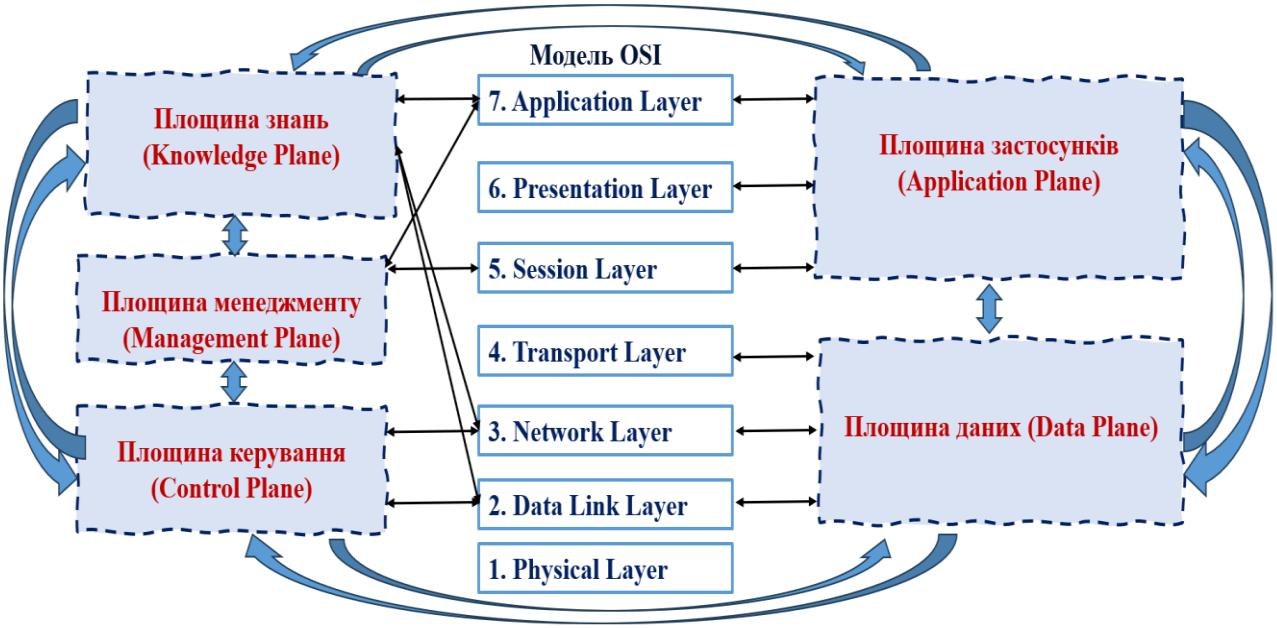

Рис. 2. Інтелектуально-конфігурована мережа (KDN) та модель OSI

У межах розвитку парадигми інтелектуально-конфігурованих мереж пропонується модель інтелектуалізованої NFV, архітектура якої включає наступні основні компонент (рис. 3).

Підсистема моніторингу – збирає інформацію про стан мережі та ресурсів. Вона охоплює як пасивний моніторинг (отримання статистики трафіку, завантаження CPU/пам'яті VNFs, затримки між вузлами із систем логів або телеметрії), так і активні вимірювання (тестові пакети для перевірки пропускної здатності, часу відгуку тощо). Дані моніторингу нормалізуються і передаються до нечіткого контролера.

Нечіткий контролер – центральний елемент інтелектуалізації. Він реалізує нечітке продукційно-логічне виведення. На вхід контролера надходять поточні значення метрик, далі відбувається їхнє fuzzification (нечітке перетворення) – тобто переведення числових значень у лінгвістичні змінні. Лінгвістичні змінні та їх терми для даної предметної області визначені

експертним шляхом. Наприклад: затримка може набувати лінгвістичних значень {низька, середня, висока}; джитер {стабільний, нестабільний}; завантаження CPU VNF {низьке, помірне, високе}; пропускна здатність каналу {достатня, критична} тощо. Для кожної такої змінної задані функції приналежності (трикутні або трапецоїдні), що відображають міру належності конкретного значення метрики до того чи іншого терму.

База правил – набір нечітких продукційних правил виду "Якщо <умова>, то <дія>", які відображають експертні знання або бажану стратегію керування. Правила комбінують кілька вхідних умов. Приклади правил:

Правило 1: Якщо затримка висока і джитер нестабільний, то підвищити пріоритет трафіку реального часу (QoS) у контролері SDN.

Правило 2: Якщо затримка висока і завантаження CPU відповідного VNF високе, то збільшити ресурси для цього VNF (масштабувати вертикально або горизонтально).

Правило 3: Якщо затримка висока, але завантаження CPU низьке, то проблема не в браку ресурсів – отже, виконати перенаправлення трафіку (SDN-контролер вибере альтернативний маршрут).

Правило 4: Якщо середня використана пропускна смуга каналу наближається до межі, то розгорнути додатковий екземпляр VNF (функції обробки трафіку) на близькому до клієнтів вузлі edge.

Правило 5: Якщо показники якості обслуговування погіршуються поступово (наприклад, плавне зростання затримки і падіння пропускної здатності), то поступово збільшувати виділення ресурсів (додаючи CPU ядра або пам'ять VNF) до стабілізації.

Ці правила відображають різні сценарії. Наприклад, правила 2 і 3 разом демонструють розв'язання двоїстої ситуації: висока затримка може спричинятися або перевантаженням самого VNF (тоді треба додати йому ресурсів), або зовнішніми мережевими причинами (тоді ресурси не допоможуть – потрібна зміна маршруту). Сукупність ~10–20 таких правил покриває типові випадки керування.

Механізм нечіткого виведення – на основі заданої бази правил та поточних нечітких вхідних значень визначає, які дії слід виконати. Для цього обчислюються ступені виконання умов у кожному правилі (операції T-norm для "і", T-conorm для "або") і на їх основі формується не-чіткий набір виходів (агрегація спрацювань правил).

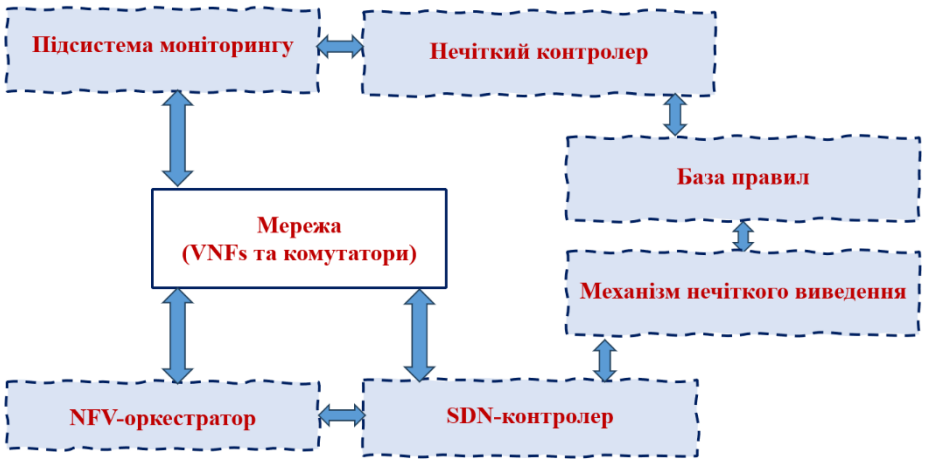

Рис. 3. Модель інтелектуалізованої NFV

Далі здійснюється defuzzification – перетворення нечіткого ріше-ння у чіткі команди. Наприклад, вихідна лінгвістична змінна може мати "необхідний рівень масштабування" з термами {незмінювати, трохи збільшити, значно збільшити}, яка після дефазифікації дає конкретне число додаткових інстанцій VNF для розгортання.

SDN-контролер та NFV-оркестратор – виконують вироблені рішення. Оркестратор NFV відповідає за керування життєвим циклом VNFs: розгортання, міграцію, виділення ресурсів гіпервізора (віртуальних CPU, пам'яті, мережевих інтерфейсів). SDN-контролер, у свою чергу, керує мережевими потоками: встановлює правила комутації на комутаторах через південний API (наприклад, OpenFlow), забезпечує перенаправлення трафіку відповідно до вимог. У типовій архітектурі ETSI NFV ці компоненти інтегровані у систему MANO (Management and Orc-hestration)
.

У даній моделі нечіткий контролер може розглядатися як розумний агент, що надбудовується над MANO: він аналізує дані моніторингу та через відкриті інтерфейси подає команди оркестратору і контролеру. У цій схемі важливою є зворотна петля: після виконання дій (наприклад, розгортання нового VNF або переключення маршруту) мережа переходить у новий стан, який знову ж відслідковується моніторингом і подається на вхід нечіткого контролера. Таким чином, реалізується цикл самоналагодження, подібний до концепції autonomic networking.

Відмітимо, що запропонована модель є концептуальною, тобто універсальною з точки зору конкретних технологій. Вона може бути реалізована на різних платформах: наприклад, як модуль для існуючих SDN-контролерів (ONOS, OpenDaylight) і NFV-менеджерів (OpenStack, Kubernetes з функціями NFV). Головне – наявність API для збору метрик і виконання дій. Сучасні SDN-рішення підтримують необхідні інтерфейси, а NFV-оркестратори накшталт ETSI OSM дозволяють через VNFM (менеджер VNF) скриптувати розгортання нових instantiations VNF. Це створює передумови для реалізації моделі на практиці.

Алгоритм роботи запропонованої системи можна описати як послідовність кроків циклічного процесу керування (схожого на цикл Моніторинг-Аналіз-План-Виконання, MAPE, у автономних системах):

1. Моніторинг (Observe): на кожному такті t система збирає значення ключових метрик. Приклади: середня затримка пакета між обраними вузлами (з вимірів ping), дисперсія затримки (джитер), відсоток втрати пакетів, середнє завантаження CPU і пам'яті на кожному VNF, обсяг трафіку на інтерфейсах. Дані можуть надходити з різних джерел – агентів на серверах, NetFlow/sFlow статистики, активних вимірювань. Всі вони агрегуються у поточний вектор стану $\mathbf{M}(t)$.

2. Аналіз/нечітке введення (Analyze): вектор $\mathbf{M}(t)$ подається на вхід нечіткого контролера. На цьому кроці виконується fuzzification: кожне числове значення метрики перекладається мовою термів. Наприклад, затримка 35 мс може дати ступінь приналежності 0.7 до терму "низька" і 0.3 до "середня" залежно від функцій приналежності. Виходом є набір нечітких значень для всіх вхідних змінних.

3. Продукційно-логічне виведення (Decide) [9]: механізм нечіткого виведення перевіряє всі правила з бази. Для кожного правила обчислюється ступінь істинності його умови на основі вхідних нечітких значень (операції мінімум/максимум для AND/OR). Далі ці значення використовуються для отримання нечітких рекомендацій по вихідних змінних. Якщо кілька правил формують різні рекомендації, вони об'єднуються (агрегація по максимуму). Результатом є нечіткі величини виходів – наприклад, "масштабувати = трохи", "змінити маршрут = так з певною впевненістю 0.5".

4. Дефазифікація і планування (Plan) [10]: нечіткі виходи перетворюються у конкретні дії. Для кількісних виходів застосовується метод центру ваги або середнього максимуму для обчислення числа. Для бінарних рішень застосовується поріг або порівняння впевненостей.

5. Виконання (Act): отриманий план транслюється у команди для NFV-оркестратора і SDN-контролера. Оркестратор через NFV-MANO API запускає або зупиняє VNFs, масштабуючи їх вертикально (виділяючи більше vCPU/пам'яті) або горизонтально (додаткові екземпляри). SDN-контролер через південні API (OpenFlow та ін.) змінює правила маршрутизації: може перевстановити маршрути для певних потоків, змінити пріоритет черг на комутаторах, реалізувати політику QoS. Такі зміни впроваджуються у мережу.

6. Оцінка (Evaluate): після паузи Δt система переходить до нового циклу моніторингу, вимірюючи, чи покращилися цільові показники (затримка, використання ресурсів) внаслідок виконаних дій. Якщо так – добре, якщо ні або стало гірше, нечітка система може спрацювати знов і запропонувати інші коригування. Таким чином, алгоритм діє ітеративно, поки мережеві показники не стабілізуються у бажаному діапазоні.

Алгоритм забезпечує адаптивність, при спокійному стані мережі дії будуть мінімальними (система фактично перебуває у моніторинговому режимі). При виникненні відхилень – цикл швидко видає керуючі впливи. На відміну від чіткого, нечіткий алгоритм може приймати рішення навіть у ситуаціях, не передбачених явно, шляхом інтерполяції правил. До того ж, він дозволяє уникнути різких дискретних змін: наприклад, якщо затримка трохи перевищила поріг, замість негайного запуску безлічі нових VNFs (що могло б статися у системі з жорсткими правилами), нечіткий контролер може видати часткову рекомендацію – збільшити ресурси ненабагато, подивитися на ефект і т. д. Це запобігає ефекту коливань і перевитрати ресурсів.

Для перевірки працездатності запропонованої моделі було розгорнуто спрощене тестове середовище. Воно складається з емульованої мережі на базі Mininet (для моделювання топології комутаторів і хостів) та контролера ONOS, інтегрованого з прототипом нечіткого контролера. В ролі NFV-інфраструктури використано кілька віртуальних машин (KVM/QEMU) з оркестратором OpenStack, на яких розгорнуто віртуальні мережеві функції: простий маршрутизатор (VNF-R) і сервер додатку (VNF-App). Нечіткий контролер реалізовано як окремий модуль, що отримує метрики через REST API від ONOS (стан мережі) та через телеметрію OpenStack (стан VNFs).

Сценарій 1: динамічне навантаження. На VNF-App генерувався змінний трафік (імітація користувачів, що приходять і йдуть) за допомогою утиліти iPerf. Протягом першої хвилини трафік зростав до насичення каналу 1 Гбіт/с. Нечіткий контролер зафіксував збільшення використання смуги до критичного рівня і спрацював: було автоматично розгорнуто другий екземпляр VNF-App на іншому сервері, а SDN-контролер поділив вхідний потік між двома екземплярами (реалізувавши балансування навантаження на рівні мережі). В результаті пропускна здатність системи подвоїлася і затримка перестала зростати. Для порівняння, стандартна система авто-масштабування OpenStack Heat із фіксованим порогом CPU 80% спрацювала із затримкою близько 10 с і теж запустила другий інстанс, але за цей час черга на першому VNF виросла і спостерігався сплеск затримки до 120 мс. Нечіткий контролер спрацював на кілька секунд раніше і підтримував затримку < 80 мс. Вимірювання iPerf показали, що сумарна пропускна здатність після масштабування досягла ~1.8 Гбіт/с (близько теоретичного максимуму з урахуванням накладних витрат протоколів), тоді як без масштабування вона обмежувалася ~940 Мбіт/с на одному екземплярі (що відповідає лімітам 1-гігабітного інтерфейсу).

Сценарій 2: деградація каналу. У другому експерименті штучно збільшувалася затримка і втрати на одному із сегментів мережі (між двома комутаторами), щоб імітувати погіршення стану лінії. При цьому трафік реального часу (UDP-потік) передавався через цей сегмент. Нечітка система, отримавши високі значення затримки та втрат, виконала правила, подібні до C і D: SDN-контролер перенаправив критичний трафік альтернативним маршрутом (через резервний шлях з меншою затримкою), а пріоритет для нього було підвищено. В результаті втрати пактів у потоці знизилися з 5% до <1%, а джитер стабілізувався. Одночасно нефоновий трафік, менш критичний, залишався на старому маршруті. Таким чином, якість сервісу для важливого потоку була збережена. Без нашої системи, в режимі звичайного протоколу маршрутизації OSPF, мережа не переналаштувалася б аж поки лінк не впав би повністю або поки адміністратор не втрутився б і весь трафік продовжував би страждати від втрат.

Сценарій 3: стабільний період. Коли навантаження невелике і всі метрики у зеленій зоні, система за правилом E зменшує активність моніторингу. Ми спостерігали, що при стабільній роботі протягом > 5 хв частота активних вимірювань (ping, iPerf-тестів) скоротилася вдвічі, що знизило накладний трафік приблизно на 3%. Як тільки почали вводити навантаження або штучно змінювати затримку, частота тестів повернулась до початкової – контролер автоматично підлаштував цей параметр.

Загалом, експерименти підтвердили працездатність концепції та моделі. Система на базі нечіткої логіки коректно інтерпретує комбінації метрик і виконує доцільні дії. Важливо, що в жодному сценарії запропонований підхід не погіршив ситуацію (що є ризиком при непра-

вильних спрацюваннях): завдяки поступовості впливу та багатокритеріальності рішень не було зафіксовано зайвого масштабування або "миготіння" перемикань. Отримані результати узгоджуються з висновками інших дослідників щодо ефективності нечітких методів: зокрема, зменшення використання ресурсів при тому ж навантаженні, що демонстрували Silva та ін. у своєму алгоритмі, та уникнення різких стрибків CPU, властивих пороговим авто-скейлерам.

Висновки

Представлено модель інтелектуалізованої віртуалізації мережевих функцій на основі нечіткого продукційно-логічного виведення, призначену для середовищ програмно-конфігурованих мереж. У рамках дослідження сформовано архітектуру системи, що поєднує моніторинг мережевих метрик, нечіткий контролер рішень та засоби оркестрації SDN/NFV. Впровадження нечіткого логічного виведення у процес керування NFV дозволяє досягти самоадаптації мережевої інфраструктури до змінних умов. Переваги підходу підтверджено експериментально: модель реагує на перевантаження швидше та точніше, ніж традиційні порогові алгоритми, що веде до покращення показників якості обслуговування і оптимізації використання ресурсів. Зокрема, показано зниження пікових затримок і варіабельності, запобігання втратам пакетів за рахунок вчасного масштабування або перебалансування трафіку. Нечіткий підхід дає змогу уникнути недоліків жорстких схем – система не перемикається хаотично між станами, а плавно регулює параметри, враховуючи кілька факторів одночасно. Це сприяє підвищенню стійкості мережі до навантажень.

Отже, отримані результати підтверджують ефективність інтеграції технологій SDN/NFV із засобами штучного інтелекту на базі нечіткої логіки. Запропонована модель може стати основою для побудови нових систем керування мережами наступного покоління, які будуть самокерованими, гнучкими та надійними. У подальших дослідженнях планується розширити базу правил для врахування енергоспоживання та вартості оренди хмарних ресурсів, а також випробувати модель у розподіленому середовищі для автоматичного налаштування параметрів нечіткої системи під конкретну мережу. Це дозволить ще більше підвищити рівень інтелектуалізації мережевої інфраструктури, відкриваючи шлях до повністю автономних мереж.

Список літератури

1. Cisco. What is software-defined networking (SDN)? [Електронний ресурс]. – Режим доступу: https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html – Дата доступу: 23.03.2025.

2. ThousandEyes. NFV — Network Functions Virtualization. [Електронний ресурс]. – Режим доступу: https://www.thousandeyes.com/learning/glossary/nfv-network-functions-virtualization – Дата доступу: 23.03.2025.

3. Ahmed T. et al. Wireless Networks Performance Monitoring Based on Passive-active QoS Measurements // Int. J. of Computer Networks & Communications, 12(1), 2020.

4. Solozábal F. et al. Virtual network function placement optimization with deep reinforcement learning // IEEE J. Sel. Areas Commun., 38(2), 2020 – pp. 292-303.

5. Kravchenko Y., Herasymenko K., Starkova O., Bulgakova A. Routing Technology Based On Virtualization Software-Defined Networking Concepti. CEUR Workshop Proceedings, 2024, 3806, pp. 226–238.

6. Бородай Д., Кравченко Ю. Концептуальна модель інтелектуалізованого керування мережними ресурсами за реалізації парадигми SDN/NFV. Сучасні інформаційні технології. 2024, №1(3), С. 82–87

7. Comprehensive Survey on Knowledge-Defined Networking. MDPI : website. URL: https://www.mdpi.com/2673-4001/4/3/25 – Дата доступу: 28.03.2025.

8. Заць О., Стрілець В., Шматков С., Ющенко В. (2024). Віртуалізація мереж як підхід до оптимізації комп'ютерних мереж. Вісник Харківського національного університету імені

В. Н. Каразіна, Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління» , 61 , 33-43. https://doi.org/10.26565/2304-6201-2024-61-04

9. Liu X. et al. Fuzzy Auto-Scaler for adaptive threshold control // Proc. IEEE/ACM CCGrid 2017. – DOI: 10.1109/CCGRID.2017.8.

10. Silva S.N. et al. Application of Fuzzy Logic for Horizontal Scaling in Kubernetes Environments within the Context of Edge Computing // Future Internet, 2024, 16(9):316.

D. Kulchytskyi, Y. Zhukov

A MODEL FOR CONTROLLING VIRTUALIZED NETWORK FUNCTIONS UNDER DYNAMIC LOAD CHANGES

The paper addresses the challenge of improving the management efficiency of virtualized network functions (NFV) in software-defined network (SDN) environments under dynamic load changes and varying Quality of Service (QoS) requirements. The increasing scale and complexity of network structures necessitate the introduction of intelligent automation mechanisms for network infrastructure management.

A conceptual model of intelligent NFV is proposed, integrating fuzzy production-rule inference ("if-then" rules) into the orchestration process of virtual network functions. This model enables real-time monitoring of network metrics and resource states to automatically make decisions regarding scaling, resource allocation, and routing, thereby adapting network operations to current conditions.

It is argued that traditional approaches with rigid threshold settings or static policies cannot provide sufficient flexibility and responsiveness in modern networks. In contrast, the use of fuzzy logic allows for processing imprecise input data, incorporating expert knowledge, and achieving smooth adaptive management without abrupt transitions. The paper outlines the structure of the proposed model, describes its operating algorithm, and provides examples of fuzzy rules.

The application of this approach demonstrates more efficient use of data center resources, maintains stable latency and other QoS metrics even under sudden traffic changes, and reduces the risk of network overload. This is particularly important for critical services requiring continuous operation and low latency (real-time services such as VoIP, video streaming, IoT, etc.).

Keywords: software-defined networks; network function virtualization; fuzzy production-rule inference; orchestration; quality of service (QoS); intelligent control system.